TEKNILLINEN KORKEAKOULU
Sähkö- ja tietoliikennetekniikan osasto

Anssi Kärkkäinen

# Menetelmiä radiotaajuisen mittausaineiston analyysiin ja visualisointiin

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi diplomi-insinöörin tutkintoa varten Espoossa 27. 1. 2005

Työn valvoja — Professori Olli Simula

Työn ohjaaja — DI Liisa Terho

TEKNILLINEN KORKEAKOULU Diplomityön tiivistelmä


Tekijä: Anssi Kärkkäinen


Työn nimi: Menetelmiä radiotaajuisen mittausaineiston analyysiin ja visualisointiin


Päivämäärä: 25.1.2005 Sivumäärä: 68



Osasto: Sähkö- ja tietoliikennetekniikan osasto

Professuuri: T-115 Informaatiotekniikka

Työn valvoja: Professori Olli Simula

Työn ohjaaja: DI Liisa Terho



Tiivistelmä:

Sähkömagneettisen spektrin käyttö on lisääntynyt jatkuvasti ja samalla signaalitiheys on kasvanut suureksi. Spektrin valvonnan sekä taajuushallinnan ja –monitoroinnin parantamiseksi on lisääntynyt kiinnostusta automatisoida signaalien tunnistamista ja luokittelua, ja siten nopeuttaa operaattorin päätöksentekoa. Sovellustarpeita on sekä siviili- että sotilasjärjestelmissä.

Informaatiotekniikan menetelmät tarjoavat keinoja analysoida ja visualisoida sähkömagneettisesta spektristä kerättyä mittausaineistoa. Menetelmiä voidaan käyttää hahmontunnistusjärjestelmän suunnittelussa. Tässä työssä tutkittuja menetelmiä ovat: itseorganisoituva kartta, pääkomponenttianalyysi, oppiva vektorikvantisointi, lähimmän naapurin menetelmä ja monikerrosperseptroni-neuroverkko. Työn tarkoituksena on tutkia ja demonstroida edellä mainittuja menetelmiä analyysijärjestelmän suunnittelussa käyttäen sovelluskohteena radiotaajuista mittausaineistoa. Lisäksi tarkoituksena oli tuottaa ohjelmakoodia, jota voidaan soveltaa jatkossa eri havaintoaineistolle.

Piirteiden valinta mittausaineistosta on oleellinen osa hahmontunnistusjärjestelmässä. Signaalinäytteistä muodostettiin piirteitä kahdella tavalla: aika- tai taajuusesitystä hyödyntäen ja aallokemuunnosta soveltaen. Muodostetut piirteet toimivat syötejoukkona yllä mainituille menetelmille.

Tulokset osoittavat, että informaatiotekniikan menetelmiä voidaan käyttää järjestelmän suunnittelun eri vaiheissa. Itseorganisoituvan kartan avulla voidaan tarkastella ja visualisoida piirteitä sekä myös luokitella niitä. Monikerrosperseptroni-verkkoa ja lähimmän naapurin menetelmää voidaan käyttää radiotaajuisten signaalien luokitteluun. Jatkossa on kehitettävä ja tutkittava piirteiden laskentaan käytettyjä menetelmiä, koska tässä työssä käytetyt menetelmät eivät tuota riittävän hyvää tulosta ajatellen taajuushallinnan ja –monitoroinnin automatisointia.



Avainsanat:
informaatiotekniikan menetelmät, itseorganisoituva kartta, piirteen muodostus, radiotaajuinen, aallokemuunnos

HELSINKI UNIVERSITY ON TECHNOLOGY Abstract of the Master's Thesis


| | |
|---|---|
| Author: | Anssi Kärkkäinen |
| Name of the Thesis: | Methods for Analysis and Visualization of Radio Frequency Measurement Data |
| Date: | 25.1.2005 Number of pages: 68 |
| Department: | Department of Electrical and Communications Engineering |
| Professorship: | T-115 Computer and Information Science |
| Supervisor: | Professor Olli Simula |
| Instructor: | M.Sc. Liisa Terho |



Abstract:

Use of electromagnetic spectrum has increased continuously and at the same time the signal density has become large. There has been a lot of interest to automate signal recognition and classification and thus provide faster decisions. Both military and civilian applications have been under investigation.

Methods of information science provide techniques for analysis and visualization of measurement data which is collected from electromagnetic spectrum in this case. These methods can be used in designing of new pattern recognition systems. The methods, studied in this thesis, are: the self-organizing map, principal component analysis, learning vector quantization, nearest neighbor method and multilayer perceptron neural network. The intention of the thesis was to examine and demonstrate these methods in designing of analysis system when test data is radio frequency measurement data. In addition to this, the goal was to produce software code for future use.

Choosing features is a very essential part of pattern recognition system. Features were generated from the measurement data in two different ways. In the first method time or frequency domain was used and the other method was based on the Wavelet transform. Produced features were set as input data for the methods of information science mentioned above.

The results show that these analysis methods can be used in different phase of designing the system. With the self-organizing map we can study and visualize features and also classify them. Multilayer perceptron neural network and the nearest neighbor method are useful of classification of radio frequency measurement data. In the future, the methods used for feature generation need more examining and development, because the methods used in this thesis do not give results good enough when we think about automatic frequency control and monitoring systems.



Keywords:
methods of information science, the self-organizing map, feature generation, radio frequency, wavelet transform

# Alkulause

Tämä diplomityö on tehty Riihimäellä sijaitsevan Puolustusvoimien Teknillisen Tutkimuslaitoksen Elektroniikka- ja informaatiotekniikkaosastolle.

Diplomityön tekeminen ei olisi ollut mahdollista ilman työnantajan joustavaa ja ymmärtäväistä suhtautumista opiskeluun. Haluan kiittää kaikkia niitä henkilöitä, jotka ovat olleet edesauttamassa työn tekemisessä. Erityisesti haluan kiittää työn ohjaajaa diplomi-insinööri Liisa Terhoa arvokkaista kommenteista ja ideoista, joiden avulla tutkimus voitiin saattaa haluttuun suuntaan ja lopputulokseen. Kiitän myös työn valvojaa professori Olli Simulaa korjausehdotuksista mahdollisimman hyvän diplomityön tuottamiseksi.

Lopuksi suuri kiitos perheelleni ja erityisesti vaimolleni Sannalle, jonka myötämielinen suhtautuminen työn ohessa opiskeluun on mahdollistanut diplomi-insinööriopinnot. Ilman hänen ymmärtäväistä asennettaan työ ei olisi koskaan tullut valmiiksi.

Hyvinkäällä 27. tammikuuta 2005

Anssi Kärkkäinen

# Sisällysluettelo

# Lyhenteet

| | |
|---|---|
| AM | amplitudimodulaatio, Amplitude Modulation |
| ASK | amplitudisiirtoavainnus, Amplitude Shift Keying |
| A1 | kantoaaltomodulaation (CW) laji |
| BMU | parhaiten sopiva yksikkö, Best Matching Unit |
| CW | kantoaaltomodulaatio, Carrier Wave |
| CWT | jatkuva aallokemuunnos, Continuous Wavelet Transform |
| DFT | diskreetti Fourier-muunnos, Discrete Fourier Transform |
| DSB | kaksisivukaistainen modulaatio, Double Side Band |
| FM | taajuusmodulaatio, Frequency Modulation |
| FSK | taajuussiirtoavainnus, Frequency Shift Keying |
| GLONASS | satelliittinavigointijärjestelmä, Global Navigation Satellite System |
| GPS | satelliittipaikannusjärjestelmä, Global Positioning System |
| HF | korkeat taajuudet, High Frequency |
| kNN | k:n lähimmän naapurin menetelmä, k-Nearest Neighbor method |
| LEF | matalaenergiakehys, Low Energy Frame |
| LOG | logaritminen modulaatiolaji |
| LSB | alempi sivukaista, Lower Side Band |
| LVQ | oppiva vektorikvantisointi, Learning Vector Quantizisation |
| MASK | monitilainen amplitudisiirtoavainnus, Multiple Amplitude Shift Keying |
| MLP | monikerrosperseptroni, Multilayer Perceptron |
| PCA | pääkomponenttianalyysi, Principal Component Analysis |
| PSK | vaihesiirtoavainnus, Phase Shift Keying |
| RF | radiotaajuinen, Radio Frequency |
| RMS | tehollisarvo, Root Mean Square |
| SC | spektrivuo, Spectral Flux |
| SHF | ylikorkeat taajuudet, Super High Frequency |
| SOM | itseorganisoituva kartta, Self-Organizing Map |
| SR | spektrin putoamispiste, Spectral Roll-off point |
| SSB | yksisivukaistainen modulaatio, Single Side Band |
| UHF | ultrakorkeat taajuudet, Ultra High Frequency |
| USB | ylempi sivukaista, Upper Side Band |

| | |
|---|---|
| VHF | hyvin korkeat taajuudet, Very High Frequency |
| WLAN | langaton lähiverkko, Wireless Local Area Network |
| WPD | aallokepakettihajotelma, Wavelet Packet Decomposition |
| ZC | nolla-ylitys –nopeus, Zero-Crossing rate |

# 1 Johdanto

Sähkömagneettisesta spektristä kerätyn informaation analysointimenetelmät on kiinnostava tutkimusaihe sekä siviili- että sotilassovellusten näkökulmasta. Siviilipuolen sovelluskohteita ovat esimerkiksi taajuushallinta ja sähkömagneettisen spektrin käytön valvonta luvattomien lähettimien löytämiseksi. Toisaalta spektrin analysoimisella ja valvonnalla voidaan löytää ja paikantaa tahattomat häiriölähteet kuten vialliset lähettimet.

Tulevaisuudessa signaalinkäsittelyjärjestelmät ovat yhä oleellisempi osa myös sotatekniisiä järjestelmiä. Signaalitiheys spektrissä on kasvanut jatkuvasti, joten automatisoinnin tarve on lisääntynyt yhä enemmän. Sotatekniikan sovelluksissa signaalinkäsittelyn tarpeet liittyvät usein automaation lisäämiseen ilmaisun, analyysin, luokittelun ja tunnistamisen alueella monimutkaisessa signaaliympäristössä. Lisääntynyt analyysikyky, jossa signaalinkäsittelyllä on merkittävä osuus, nostaa myös automaatioastetta ja muuttaa operaattorin tehtävän luonnetta rutiinien suorittajasta päätöksentekijäksi.

Signaalianalyysin tavoitteena on havaita, luokitella, tunnistaa ja mahdollisesti jopa yksilöidä lähetteitä mitatusta aineistosta. Tulevaisuudessa reaaliaikaisen signaalianalyysin oletetaan perustuvan yhä enemmän ohjelmistopohjaisiin sovelluksiin. Toisaalta myös tallenteiden ei-reaaliaikaisen analysoinnin automatisointiin liittyy tunnistusmenetelmien kehittäminen luokittelun ja hahmontunnistusparadigman pohjalta. Tutkimus tuottaa perusteita arvioida ratkaisumalleja, algoritmeja ja rakenteita sekä kehittää toiminnallisia demonstraattoreita.

Perinteisessä signaalianalyysissa pyritään tunnistamaan lähete teknisten piirteiden avulla. Yksinkertaisimmillaan signaalianalyysi voi olla lähetteen taajuuden ja voimakkuuden havainnointia aikaan sidottuna. Sähkömagneettisesta spektristä kerätyn moniulotteisen datan tarkempi analysointi edellyttää informaatiotekniikan keinoja. Tiedonlouhinnan ja hahmontunnistamisen metodeja pyritään soveltamaan taajuushallinnan ja -monitoroinnin keräämiin tietokantoihin, tiedustelutietoihin ja mittausaineistoon. Uusia analyysimenetelmiä kuten esimerkiksi neuroverkkoja ja geneettisiä algoritmeja tul-

laan soveltamaan yhä enemmän kohteen tunnistukseen. Signaaliympäristön monimutkaiset rakenteet vaativat tehokkaita tuloksien visualisointimenetelmiä.[1]

Tässä työssä analysoidaan eräiden menetelmien soveltuvuutta signaalianalyysijärjestelmän kehittämiseen käyttäen testiaineistona sähkömagneettisesta spektristä kerättyä mittausaineistoa. Päämääränä on selvittää, miten valitut informaatiotekniikan menetelmät soveltuvat radiotaajuisen mittausdatan hahmottamiseen, visualisointiin ja luokitteluun analyysin eri vaiheissa. Rajatusta mittausaineistosta johtuen työn tavoitteena on havainnollistaa informaatiotekniikan menetelmien käyttöä analyysimenetelmien ja -ohjelmistojen jatkokehittämisprosessissa sen sijaan, että pyrittäisiin parhaaseen mahdolliseen tulokseen hahmontunnistuksessa ja luokittelussa. Tavoitteena on myös tuottaa perustyökaluja Matlab-sovellukseen vastaavia analyyseja varten. Sovellettuja informaatiotekniikan menetelmiä ovat itseorganisoituva kartta, pääkomponenttianalyysi, lähimmän naapurin menetelmä, oppiva vektorikvantisointi ja monikerrosperseptroni-neuroverkko.

Työn pääpaino on itseorganisoituvan kartan soveltamisessa. Professori Teuvo Kohosen kehittämä itseorganisoituva kartta (self-organizing map, SOM) on yksi suosituimmista neuroverkkojen malleista. SOM-algoritmi perustuu ohjaamattomaan kilpailuoppimiseen, joka tarkoittaa sitä, että verkon opettaminen on syötedatan aikaansaamaa ja verkon neuronit kilpailevat toistensa kanssa. Ohjatun oppimisen algoritmit, kuten lähimmän naapurin menetelmä, oppiva vektorikvantisointi ja monikerrosperseptroni, vaativat kunkin syötevektorin tavoitearvojen tai luokan tuntemista, mutta SOM:lla tätä vaatimusta ei ole. Itseorganisoituvaa karttaa on sovellettu laajalla alueella konenäöstä tekstianalyysiin ja prosessiohjauksesta neuropsykologiseen tutkimukseen.[2]

Tämän työn päätutkimusongelma on itseorganisoituvan kartan soveltuvuus radiotaajuisen mittausaineiston luokitteluun tarkoitetun hahmontunnistusjärjestelmän kehittämisen eri vaiheisiin. Valittuja menetelmiä tutkitaan kehittämällä ja soveltamalla Matlab-työkaluja etukäteen taltioituun datajoukkoon. Muita tutkimusongelmia ovat:

Millaisia menetelmiä voidaan käyttää piirteiden muodostamiseen radiotaajuisesta havaintoaineistosta?

Miten luokitteluun käytettäviä piirteitä voidaan visualisoida?

Miten oppivaa vektorikvantisointia voidaan käyttää SOM:n klusteroinnin parantamiseen?

Miten lähimmän naapurin menetelmää ja monikerrosperseptroni -neuroverkkoa voidaan soveltaa luokitteluun?

Miten PCA soveltuu havaintoaineistosta lasketun piirrematriisin dimension pienentämiseen?

Mittausaineistosta irrotettujen piirteiden hyvyyttä ei testata matemaattisilla menetelmillä. Piirteitä lasketaan eri tavoilla, mutta niiden keskinäistä paremmuutta ei analysoida erikseen. Työssä käytettyjen piirrealgoritmien hyvyyttä on tutkittu aikaisemmissa tutkimuksissa erilaisilla luokittimilla, joten tässä työssä testataan ja havainnollistetaan visualisointivaihetta piirrevalinta-algoritmin osana. Tarkasteltujen menetelmien laskennallista tehokkuutta ei tutkita tässä diplomityössä. Menetelmien monimutkaisuutta ja ohjelmoinnin hankaluutta ei myöskään tarkastella. Muistikapasiteetin tarpeen tutkiminen rajataan tämän työn ulkopuolelle.

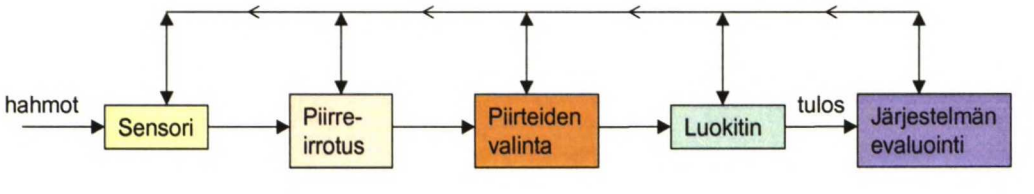
Ensimmäinen luku on johdanto, jossa selvitetään työn yleinen viitekehys, tutkimusongelma, rajaukset sekä esitellään työn sisältö. Toisessa luvussa käsitellään hahmontunnistamista yleisellä tasolla sekä perehdytään hahmontunnistuksen menetelmiin. Kolmas luku käsittelee itseorganisoituvaa karttaa ja neljäs luku riippumattomien komponenttien analyysia. Viidennessä luvussa esitellään lähimmän naapurin menetelmään ja kuudes luku käsittelee oppivaa vektorikvantisointia. Seitsemännessä luvussa käsitellään monikerrosperseptroni -neuroverkkoa. Kahdeksas luku keskittyy signaalinkeruulaitteiston esittelyyn ja näytejoukkona käytetyn datan ominaisuuksiin. Yhdeksännessä luvussa esitellään valittujen piirteenmuodostusalgoritmien rakenne ja toiminta ja kymmenennessä luvussa esitetään data-analyysin tulokset. Viimeisessä luvussa esitetään yhteenveto ja johtopäätökset.

# 2 Hahmontunnistus- ja tiedon louhintamenetelmistä

## 2.1 Hahmontunnistus luokitteluprosessina

Hahmontunnistus on monivaiheinen prosessi [6], jonka tavoitteena on luokitella havainnot tiettyihin kategorioihin tai luokkiin. Teollisuuden automatisoitumien ja informaation käsittely ovat tulleet entistä tärkeämmäksi. Niinpä hahmontunnistus on nykyään oleellinen osa älykkäitä järjestelmiä, jotka on rakennettu päätöksentekoa varten. Esimerkkejä hahmontunnistuksen sovelluksista ovat konenäkö, merkkien tunnistaminen, tietokoneavusteinen diagnoosin teko ja puheentunnistaminen.

Kuvassa 1 on esitetty luokittelujärjestelmän suunnittelun vaiheet. Kuvassa vasemmalla järjestelmä havainnoi ympäristöään eli kerää sensorin avulla dataa. Kerätystä datasta muodostetaan piirteitä piirreirrotusalgoritmin avulla, minkä jälkeen muodostetuista piirteistä valitaan kaikkein oleellisimmat. Lopuksi valitut piirteet luokitellaan (tai analysoidaan valitulla tiedon louhinnan menetelmällä) ja lopputulosta evaluoidaan. Tärkeä osa järjestelmää on palauteinformaatio, jonka perusteella voidaan suunnitella järjestelmän eri vaiheet uudella tavalla ja hienosäätää prosessia siten, että järjestelmällä saavutetaan haluttuja tuloksia.




**Kuva 1.** Luokittelujärjestelmän suunnitteluun sisältyvät vaiheet.

Kuten palautenuolista nähdään, suunnitteluprosessin vaiheet eivät ole riippumattomia toisistaan. Prosessin osajärjestelmät sisältävät suhteita toisiinsa, ja riippuen lopputuloksesta suunnittelussa voidaan siirtyä takaisin edellisiin vaiheisiin, jotta kokonaissuorituskyky paranee.

Oleellista luokitteluprosessissa on piirreirrotuksen onnistuminen. Hyvä piirre erottelee erilaiset hahmot selvästi erilleen ja vastaavasti lähellä toisiaan olevilla hahmoilla on samankaltainen piirre. Piirreirrotus on ongelmariippuvainen, joten piirreirrotusalgoritmi on suunniteltava syötedatan perusteella. Toinen ongelma on luokittelussa käytettyjen piirteiden määrä. Käytännössä piirteitä generoidaan enemmän kuin on tarpeen, jotta piirrejoukosta voidaan valita optimaalinen piirrevektori luokiteltaviksi.

Kolmas haaste hahmontunnistuksessa on rakentaa laadukas luokitin tai muu hahmontunnistusjärjestelmä, joka pystyy suoriutumaan annetusta tehtävästä. Yksinkertaisimmillaan luokittimen luokitelurajapinta on suora, joka jakaa näytteet kahteen luokkaan. Monimutkaisimmissa tapauksissa tarvitaan erilaisia epälineaarisia luokittelupintoja. Luokitinjärjestelmän suunnitteluun ja valintaan liittyy myös optimointikriteeri, jonka avulla luokittimen rajapinnat pyritään rakentamaan optimaalisesti. Ongelmana onkin, millaista epälineaarisuutta tarvitaan ja minkä tyyppinen optimointikriteeri valitaan.

## 2.2 Mitä on tiedon louhinta?

Tiedon louhinta on laaja käsite [3, 4]. Yksinkertaisimmillaan se on tiedon etsimistä suuresta tietokannasta. Digitaalisen datan tuottaminen on kasvanut valtavasti, jolloin myös mielenkiinto tietokantoja kohtaan on kasvanut. Mielenkiintoisia tietovarastoja löytyy sekä tieteelliseltä että teolliselta sektorilta: bio- ja geoinformatiikasta, tähtitieteestä, ekologiasta, kombinatorisesta kemiasta, lokitietokannoista, kieliteknologiasta, kauppojen myyntitietokannoista ja teollisuusprosessien seurantatietokannoista.

Tiedon louhinta voidaan määritellä seuraavasti:

> Tiedon louhinta on usein suuren, havainnoitavan datamäärän analysointia, jonka tarkoituksena on löytää olettamattomia suhteita ja kuvata dataa uusilla menetelmillä siten, että se on ymmärrettävää ja hyödyllistä datan hyödyntäjälle.[3]

Tiedon louhintaa sovelletaan usein dataan, joka on jo kerätty. Esimerkiksi teollisuusprosessista tallennetut parametrit tai pankkien luottokorttitiedot ovat tällaista dataa. Samalla tavalla tässä työssä käytetty sähkömagneettisesta spektristä kerätty data on valmiiksi tallennettu tietokannaksi. Tiedon louhinta ei näin ole niin kutsuttu online-prosessi. Tiedon louhinnassa ei ole oleellista, miten tietoa on kerätty. Juuri tämä erottaa tiedon lou-

hinnan tilastotieteistä, jossa data on usein kerätty tietyllä strategialla halutun tuloksen aikaan saamiseksi.

Tiedonlouhinnassa kehitetään ja sovelletaan matemaattisia, tilastollisia ja ohjelmallisia menetelmiä tietovarastojen analysoimiseksi. Tavoitteena on etsiä algoritmien avulla tietokannoista kiinnostavaa tietoa, joka voi olla esimerkiksi globaali kuvaus kuten todennäköisyysjakauma, klusteroituvuus tai visuaalinen yhteenveto. Toisaalta voidaan etsiä lokaaleja ominaisuuksia kuten toistuvia hahmoja, tilastollisia riippuvuuksia tai aikasarja-aineistojen hahmoja.[4]

## 2.3 Hahmontunnistuksen ja tiedon louhinnan tehtäviä

Tiedon louhinnalla on karkeasti kaksi päätehtävää: datan kuvaaminen ja ennustaminen [4]. Näistä datan rakenteen kuvaaminen on tärkeämpi, koska se selittää dataan sisältyviä ilmiöitä. Tässä tutkimustyössä keskitytään datan hahmottamiseen ja kuvaamiseen.

Myöhemmin esitetyt hahmontunnistamisen ja tiedon louhinnan menetelmät kuvaavat, miten tietokannasta kaivetaan lisätietämystä. Hahmontunnistuksen ja tiedon louhinnan tehtävät taas kertovat, mitä halutaan oppia datasta. Usein data-analyyseissa päätetään aluksi tehtävä ja sen jälkeen valitaan sopiva menetelmä halutun tehtävän suorittamiseen. Oppimistehtäviä ovat luokittelu, tiheysestimointi, regressio ja klusterointi. Muita tehtäviä ovat yhteenveto- ja riippuvuusanalyysit.

Luokittelussa data- tai piirrevektoreita luokitellaan ominaisuuksien mukaan eri luokkiin. Päätösrajapinta jakaa data-avaruuden eri osiin ja erotusfunktio antaa vastearvon, jonka mukaan näyte saa luokkatiedon.

Tiheysestimoinnin tarkoituksena on löytää approksimaatiofunktio tuntemattomalle jakaumalle. Tiheysestimointi on yksi vaikeimmista tiedon louhinnan tehtävistä, koska kerätyssä datassa on aina puutteita johtuen reaalimaailman jatkuvuudesta. Estimaatit, kuten Gaussiset sekoitukset tai Parzen-ikkunointi ovat usein liian pelkistäviä tarkkojen tulosten saamiseksi.

Regressiolla pyritään löytämään reaaliarvoinen funktioapproksimaatio, joka saa syötteenä kohinaista dataa. Regressiota käytetään ennustamiseen.

Klusteroinnilla pyritään löytämään datasta tihentymiä annettujen kriteerien perusteella. Dataa ei ole etukäteen luokiteltu kuuluvaksi mihinkään ryhmään. Ryhmien määrää ja valintasääntöjä muuttamalla saadaan erilaisia tuloksia. Klustereiden löytäminen kuitenkin kuluttaa runsaasti laskenta tehoa. Klusteroinnissa on kolme vaihetta: datan tiheyden arviointi, klusterointi valitulla algoritmilla ja tulosten arviointi. Tiheyden arvioinnilla varmistetaan, että data ei ole satunnaisesti jakautunutta.

Yhteenvetoanalyysissa tarkastellaan moniulotteisen datan yksittäisiä muuttujia ja tehdään niiden perusteella yhteenvetoja. Kvantitatiivisella muuttujalla on yleensä tyyppi ja muita lisäarvoja kuten esimerkiksi mediaani, keskiarvo, minimi ja maksimi. Puuttuva ja epämääräinen data huomioidaan myös. Riippuvuusanalyysin avulla etsitään muuttujien välisiä riippuvuuksia. Rakenteellisella tasolla riippuvuusmalli kuvaa, mitkä muuttujat ovat riippuvia toisistaan paikallisesti tai globaalisti. Kvantitatiivinen taso taas kuvaa riippuvuuksien numeerista vahvuutta. Lineaarinen keskinäiskorrelaatio on yksi standardimetodi.

## 2.4 Luokittelumenetelmiä

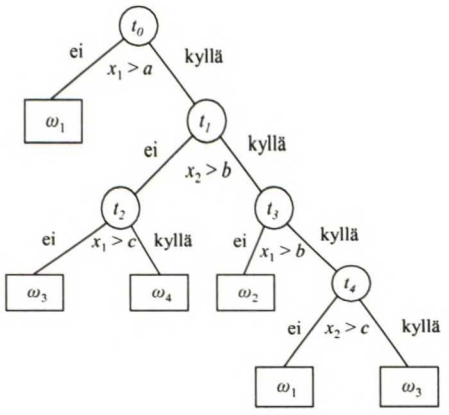
Suosittuja hahmontunnistuksen ja tiedonlouhinnan luokittelu- ja analyysimenetelmiä ovat neuraalilaskennan sovellukset ja itseorganisoituvat kartat [3, 4]. Muita käytettyjä menetelmiä ovat Bayes-verkot, päätöspuut, assosiaatio- ja sekvenssisäännöt, geneettiset algoritmit ja sumean logiikan menetelmät.

Neuraalilaskennan menetelmissä lähtökohtana on ihmisaivojen prosessointia mallintava rinnakkaislaskenta. Ihmisaivot ovat kuin monimutkainen, epälineaarinen ja rinnakkaislaskentaa käyttävä tietokone, jonka toimintaa neuraalisissa menetelmissä yritetään mallintaa neuroniverkkojen avulla. Perusyksikkönä on neuroni, jonka toiminta perustuu epälineaariseen aktivaatiofunktioon ja neuronien välisten kytkentöjen painoarvoihin.

Itseorganisoituvan kartan päätarkoitus on kuvata mielivaltaisen moniulotteinen syötedata yksi tai kaksiulotteiseksi kartaksi siten, että topologinen järjestys säilyy. Kartta koostuu neuronien muodostamasta, usein yksi- tai kaksiulotteisesta hilasta. Itseorganisoituvaa karttaa voidaan käyttää sekä luokitteluun että piirteiden etsimiseen.[5]

Bayes-verkkojen keskeinen idea on esittää tietämys hajautettuna osatotuuksien riippuvuusrakenteena, josta tarkasteltava tieto voidaan jalostaa esiin yksinkertaisilla laskutoimituksilla. Bayes-verkot ovat suunnattuja graafeja, joiden solmut edustavat satunnaismuuttujia, esimerkiksi $X_1$, $X_2$,...,$X_n$, tai tarkemmin niiden todennäköisyyksiä (tai tiheysjakaumia) saada tiettyjä arvoja, kuten $\Pr(X_1 = a)$. Verkon linkit edustavat tilastollisia riippuvuussuhteita eri satunnaismuuttujien välillä, mitkä voidaan esittää ehdollisina riippuvuuksina, esimerkiksi solmujen $A$ ja $B$ välillä muodossa $\Pr(X_B|X_A)$.

Päätöspuut ovat moniportaisia päätössysteemejä, joissa luokat vaihe kerrallaan pilkotaan, kunnes saavutetaan lopullinen hyväksyttävä luokka. Lopussa piirreavaruus on jaettu yksittäisiin alueisiin, jotka vastaavat kutakin luokkaa. Luokka, johon piirrevektori sijoitetaan, määräytyy vaiheittain tehtävistä päätöksistä päätöspuun solmujen muodostamalla polulla [6]. Kuvassa 2 on esimerkki päätöspuusta. Ympyrät ovat solmuja ($t_n$), joissa päätös tehdään ja laatikot lopullisia luokkia ($\omega_n$). Muuttuja $x_n$ on piirrevektorin alkio.




**Kuva 2.** Esimerkki päätöspuusta.

Geneettisten algoritmien taustalla on Darwinin esittämä luonnollisen valinnan mekanismi. Algoritmit käyttävät sellaisia operaattoreita populaation eli joukon ongelmien ratkaisuun, että uusi joukko on aina parempi kuin edellinen etukäteen määritellyn kriteerifunktion mukaan. Prosessi perustuu ennalta valittuun iteraatiokierrosten määrään. Algoritmi tuottaa parhaan ratkaisun yleensä viimeisen populaation avulla. Jossain tapauksissa paras ratkaisu voi löytyä myös algoritmin evoluution aikana.[6]

Sumeaa logiikkaa on sovellettu useille eri hahmontunnistuksen alueille, kuten kielelliseen hahmontunnistukseen, kirjainten tunnistamiseen (kaupallisesti saatavana on mm. elektroninen muistikirja, joka tunnistaa käsin kirjoitettuja merkkejä), kuvien sisällön kuvailuun, pintojen luokitteluun sekä lääketieteelliseen ja teknilliseen diagnostiikkaan. Sumeiden systeemien teorian perustana on sumea joukko-oppi. Perinteisessä joukko-opissa objektit joko kuuluvat tai eivät kuulu annettuun joukkoon, mutta sumeassa joukko-opissa objekti voi kuulua joukkoon vain osittain. Esimerkiksi harmaa objekti kuuluu vain osittain sekä mustien että valkoisten objektien joukkoon.

Vaikka sumean logiikan teoria perustuu siis moniarvologiikan ajattelutapaan, eli totuusarvoja on enemmän kuin kaksi (eli muitakin kuin tosi ja epätosi), sen tutkimuskohde ja metodit poikkeavat monilta osin muista logiikoista. Yksi merkittävä ero on se, että sumea logiikka luopuu perinteisestä pyrkimyksestä täsmälliseen esitystapaan ja näin ollen hyväksyy selvästi epätäsmällisyyden mukanaolon.[7]

# 3 Itseorganisoituva kartta

Itseorganisoituva kartta (self-organizing map, SOM) [2] on yksi suosituimmista neuroverkkomalleista. Se kuuluu kilpailuoppimisverkkojen kategoriaan. Itseorganisoituva kartan toiminta perustuu ohjaamattomaan oppimiseen, joten verkon opettamisvaiheessa ei tarvita ulkopuolista apua tai ohjausta. Ainoastaan syötedatan ominaisuuksia käytetään hyväksi. Karttaa voidaan käyttää esimerkiksi luokitteluun ilman, että tiedetään mihin luokkaan syötedatan näytteet kuuluvat.

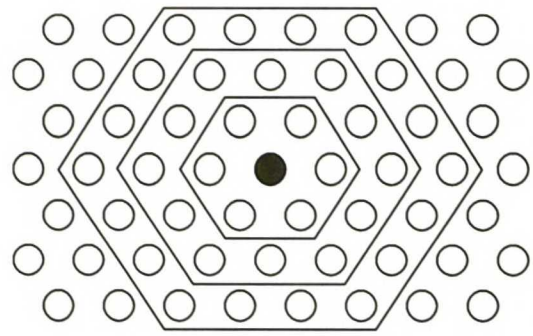
Itseorganisoituva kartta on kehitetty Teknillisen Korkeakoulun Informaatiotekniikan laboratoriossa 1980-luvun alkupuolella professori Teuvo Kohosen toimesta. Kartta on ollut erittäin käyttökelpoinen useissa erilaisissa sovelluksissa kuten teollisuusprosessien valvonnassa, matkaviestiverkon tukiasemadatan monitoroinnissa ja kuvien hakemisessa tietokannasta.

Käytännössä SOM on skaalausmenetelmä, joka projisoi moniulotteisen syötedatan pieniulotteiseen vasteavaruuteen. Kartta pyrkii säilyttämään syötedatan topologian, mikä tarkoittaa sitä, että syötedatassa lähellä olevat pisteet ovat lähellä myös kartan hilassa. Karttayksiköt eli neuronit muodostavat yleensä kaksiulotteisen hilan, joten kuvaus suo-

ritetaan moniulotteisesta avaruudesta tasoon. Useampiulotteisia hiloja voidaan myös käyttää, mutta tällöin visualisointi on vaikeampaa.

Jokainen kartan neuroni on esitetty $n$-ulotteisena painona tai mallivektorina (codebook vector) $\mathbf{m}_i = [m_{i1}, \ldots, m_{in}]$, jossa $n$ on syötevektorin dimensio. Neuronit on kytketty viereisiin neuroneihin naapuristosuhteella, joka määrää kartan topologian tai rakenteen. Tavallisesti neuronit on kytketty toisiinsa suorakaide tai heksagonaalisen topologian avulla. Neuroneiden määrä määrittää saadun kuvauksen granulariteetin, mikä vaikuttaa kuvauksen tarkkuuteen ja yleistyskykyyn. Tarkkuus ja yleistyskyky ovat vaihtoehtoisia tavoitteita, koska parantamalla toista ominaisuutta heikennetään toista.

Vierekkäiset neuronit kuuluvat neuronin $\mathbf{m}_c$ naapuristoon $N_c$. Naapuristofunktio on ajasta riippuva eli $N_c = N_c(t)$. Kuvassa 3 on esitetty erikokoisia naapuristoja heksagonaalisessa topologiassa.



**Kuva 3.** Erikokoisia naapuristoja mustalla pisteellä merkitylle yksikölle.

## 3.1 Alustus

Neuronien määrä voidaan yleensä valita niin suureksi kuin halutaan. Kuvaus ei yleensä huomattavasti kärsi vaikka neuronien määrä lähestyy syötedatan näytteiden määrää, jos naapuristofunktio on valittu huolella. Kuitenkin, jos neuronien määrä on kymmeniä tuhansia, kartasta tulee laskennallisesti epäkäytännöllisen raskas.

Viitteessä [2] on esitetty kolme erityyppistä tapaa alustaa kartan neuronit: satunnainen alustaminen, alustus näytteiden avulla ja lineaarinen alustaminen. Satunnaisessa alustamisessa mallivektoreille eli neuronien painoille annetaan satunnaiset arvot. Satunnaista alustamista käytetään yleensä silloin, kun syötedatan ominaisuuksia ei tunneta. Näyttei-

den avulla alustamisessa näytejoukosta poimitaan satunnaisesti näytteitä mallivektoreiksi. Hyvänä puolena tässä on se, että mallivektorit ovat samassa syöteavaruudessa kuin data.

Lineaarisessa alustamisessa käytetään pääkomponenttimenetelmää mallivektorien alustamiseen. Mallivektorit alustetaan siten, että ne ovat kahden ominaisvektorin virittämässä avaruudessa. Nämä kaksi ominaisvektoria ovat syötedatan kahta suurinta ominaisarvoa vastaavat vektorit. Tällä tavalla alustettu kartta on orientoitunut syötedatan mukaan sisältäen merkittävämmän määrän energiaa. Ominaisarvot voidaan laskea käyttämällä Gram-Schmidt –menetelmää.

## 3.2 Opetusvaihe

Itseorganisoituvan kartan opettaminen on iteratiivinen prosessi ajan suhteen. Opettaminen vaatii runsaasti laskentatehoa, ja on siten aikaa kuluttava prosessi. Opetusvaiheessa syötedatasta otetaan satunnaisesti näytevektori, joilla "opetetaan" itseorganisoituvaa karttaa. Opettaminen tapahtuu kaksivaiheisesti: ensimmäisessä vaiheessa valitaan voittajaneuroni näytteen samankaltaisuuden perusteella ja toisessa vaiheessa voittajaneuronin ja sen naapuriston mallivektoreita päivitetään naapuristofunktion mukaan. Prosessia toistetaan useita kertoja.

Jokaisella opetusaskeleella valitaan satunnaisesti syötedatasta näyte, jolle lasketaan samankaltaisuuden mitta jokaisen kartan mallivektorin kanssa. Parhaiten sopivaksi mallivektoriksi (Best Matching Unit, BMU) valitaan se kartan yksikkö, jonka kanssa näytevektori on samankaltaisin. Samankaltaisuus on usein määritetty etäisyysmittana. Usein mittana käytetään euklidista mittaa. Vektorin **x** euklidinen normi lasketaan kaavalla

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} \mathbf{x}_i^2}\,, \tag{1}$$

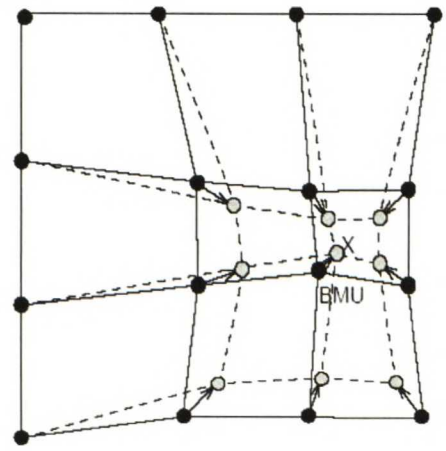
jonka perusteella voidaan laskea euklidinen etäisyys seuraavasti:

$$d_E(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|. \tag{2}$$

Muodollisesti määriteltynä parhaiten sopiva mallivektori eli BMU (merkitään $\mathbf{m}_c$) on se yksikkö, jolle

$$\|\mathbf{x} - \mathbf{m}_c\| = \min_i \{\|\mathbf{x} - \mathbf{m}_i\|\}. \quad (3)$$

BMU:n löytämiseen jälkeen seuraa mallivektoreiden eli kartan yksiköiden päivitys. Päivityksen aikana BMU siirretään hieman lähemmäs näytevektoria. Samalla siirretään myös topologiseen naapuristoon kuuluvia yksiköitä. Tällä tavalla päivitysproseduuri venyttää voittajaneuronia ja sen naapuristoa kohti näytevektoria kuten kuvassa 4 on näytetty.




**Kuva 4.** BMU:n ja sen topologisen naapuriston päivittäminen kohti näytevektoria **x**. Katkoviivalla on kuvattu tilanne päivityksen jälkeen.[8]

Parhaiten sopivan yksikön löytäminen ja mallivektorien päivittäminen vaikuttavat laskennalliseen tehokkuuteen. Jos BMU:n naapuristo on suuri, mallivektoreiden päivittämiseen kuluu paljon laskentatehoa. Tällainen tilanne saattaa olla opetusvaiheen alussa, jolloin on suositeltavaa käyttää laajaa naapuristoa. Toisaalta tilanteessa, jossa kartan neuroneja on runsaasti, laskentatehoa kuluu voittajaneuronin eli BMU:n etsimiseen. Kartan opettamiseen kuluva aika riippuu siis ohjelmiston ja tietokoneen tehokkuudesta.

Edellä kuvatun päivitysproseduurin tuloksena kartan mallivektorit kerääntyvät lähelle toisiaan siellä, missä syöteavaruuden dataa on tiheässä, ja vain muutamia mallivektoreja on siellä, missä syötedataa on harvassa. Tällä tavalla itseorganisoituvalla kartalla on taipumus approksimoida syöteavaruuden todennäköisyystiheysjakaumaa [5].

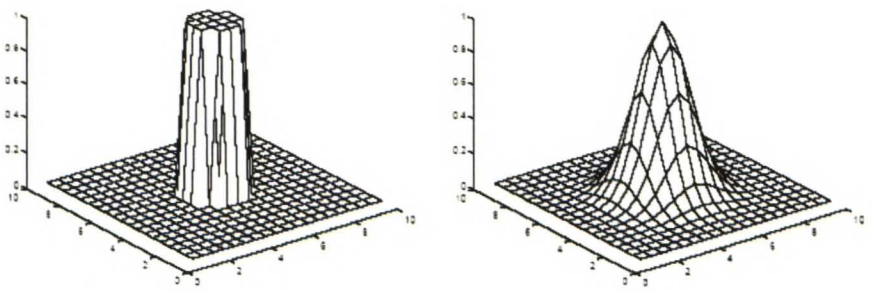
Kartan mallivektoreiden päivityssääntö on seuraava:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)], \tag{4}$$

jossa $t$ on aika. Muuttuja $\mathbf{x}(t)$ on satunnaisesti poimittu näytevektori ajan hetkellä $t$ ja $h_{ci}$ on ei-kasvava naapuristofunktio. Naapuristofunktio riippuu ajasta ja etäisyydestä eli

$$h_{ci}(t) = \alpha(t)h(d,t) \tag{5}$$

ja siinä on kaksi osaa: naapuriston muotofunktio $h(d,t)$ ja opetuskerroin $\alpha(t)$. Yksinkertainen naapuriston muoto on kupla (bubble), joka tarkoittaa, että funktio on vakio voittajayksikön naapuristossa ja muualla nolla. Toinen on gaussinen naapuristo, joka antaa hieman parempia tuloksia, mutta on laskennallisesti raskaampi. Kuvassa 5 on esitetty molemmat naapuristofunktiot.



**Kuva 5.** Kupla- (vasemmalla) ja gaussinen naapuristofunktio (oikealla).

Opetuskerroin $\alpha(t)$ on ajan mukaan vähenevä funktio. Kaksi usein käytettyä funktiota ovat lineaarinen funktio ja ajan suhteen käänteisesti verrannollinen funktio

$$\alpha(t) = \frac{A}{t+B}, \tag{6}$$

jossa $A$ ja $B$ ovat sopivia vakioita. Jälkimmäisen funktion käytöllä voidaan varmistaa, että kaikilla syötedatan näytteillä on suunnilleen samanlainen vaikutus opetustulokseen [9].

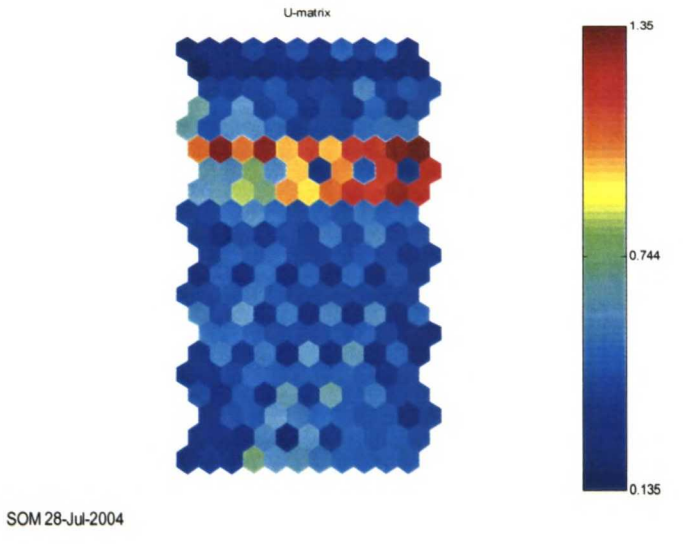
Opetus tapahtuu yleensä kahdessa vaiheessa. Aluksi käytetään suuria opetuskertoimen ja naapuriston säteen arvoja. Toisessa vaiheessa opetuskertoimen ja naapuriston säteen arvot ovat pieniä. Tällä tavalla SOM säädetään aluksi karkeasti syöteavaruuteen, ja sen jälkeen kartta hienosäädetään kohdalleen. Lineaarista funktiota käytettäessä ei tarvita

ensimmäistä vaihetta ollenkaan. Sopivien arvojen valintaan on löydetty kokeiden kautta useita "peukalosääntöjä" [2].

## 3.3 Visualisointi

Itseorganisoituva kartta on syötedatan todennäköisyystiheysfunktion approksimaatio, joten sitä voidaan käyttää datan visualisointiin. Itseorganisoituvalle kartalle on olemassa useita erilaisia visualisointitekniikoita.

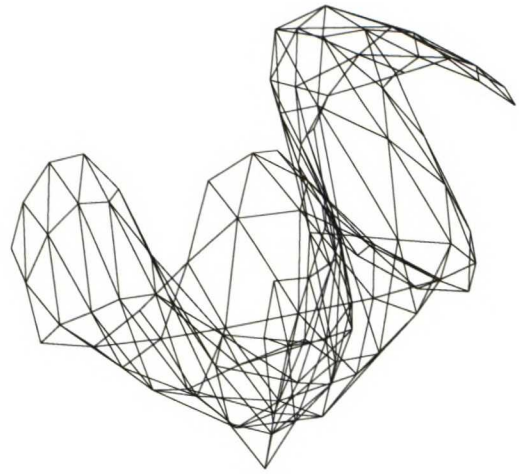
U-matriisi (Unified Distance Matrix) [10] kuvaa itseorganisoituvan kartan neuroneiden välisiä etäisyyksiä. U-matriisissa lasketaan mallivektoreiden väliset keskimääräiset etäisyydet, jotka esitetään eri väreillä solmupisteiden välillä. Tavallisesti tummat värit kuvaavat pitkiä etäisyyksiä ja vaaleat värit lyhyitä välimatkoja. Vaaleat alueet voidaan tulkita kasaantumiksi eli klustereiksi. Tummat alueet ovat klustereita erottavia alueita. Kuvassa 6 on esitetty itseorganisoituvan kartan U-matriisi. Kuvasta nähdään, että yläosa muodostaa klusterin, joka on erotettu alaosasta harvemmalla alueella (punertava alue). Kartta on muotoutunut ohjaamattoman oppimisen tuloksena. SOM:n opettaminen ja U-matriisin muodostaminen on näin ollen nopea tapa tarkastella moniulotteisen datan jakaumaa.




**Kuva 6.** Esimerkki itseorganisoituvan kartan U-matriisista.

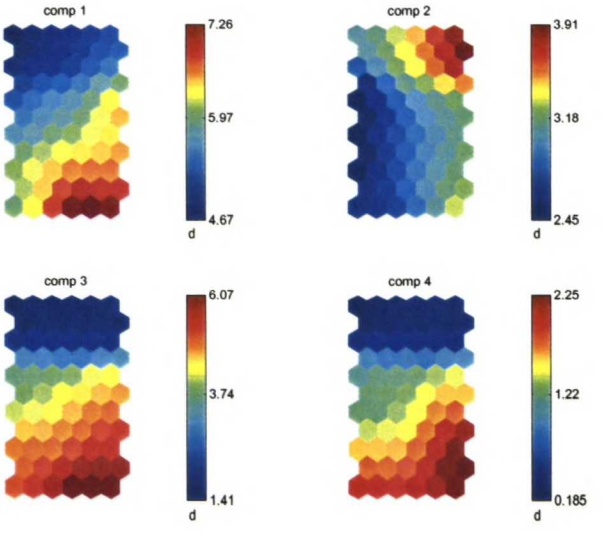
Sammonin kuvausta (Sammon's mapping) voidaan käyttää mallivektoreiden esittämiseen kaksi- tai kolmiulotteisessa avaruudessa siten, että vektoreiden välinen suhteellinen

etäisyys säilyy. Topologiset suhteet voidaan esittää piirtämällä viivat naapurivektoreiden välille, jolloin saadaan verkkomainen esitys. Sammonin kuvaus voidaan tehdä myös suoraan syötedatalle, mutta prosessi on silloin laskennallisesti raskaampi, koska itseorganisoituva kartta kvantisoi syötedatan pienempimääräiseksi mallivektorijoukoksi. Näin laskennan määrä vähenee ja prosessointitehokkuutta ei tarvita niin paljon. Kuvassa 7 on esimerkki Sammonin kuvauksesta. Verkon muoto antaa häilyvän kuvan mallivektoreiden jakaumasta.



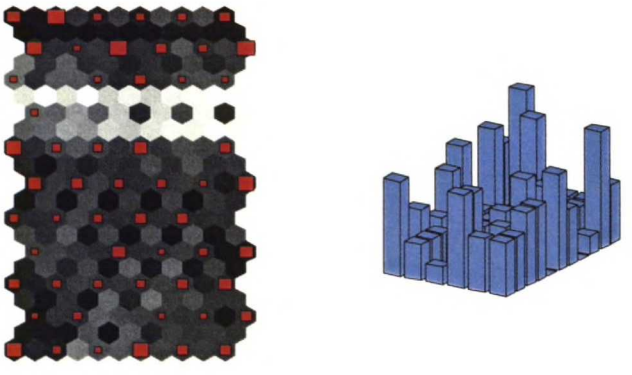
**Kuva 7.** Sammonin kuvaus.

Komponenttitasoesityksen avulla voidaan tarkastella syötedatan komponenttien suhteellisia jakaumia ja osuuksia. Komponenttitasoesityksen voidaan myös ajatella olevan viipaloitu versio itseorganisoituvasta kartasta. Jokainen komponenttitaso sisältää yhden näytevektorin komponentin suhteellisen jakauman. Tavallisesti tummat arvot kuvaavat pieniä arvoja ja vaaleat arvot suurta esiintymistiheyttä. Vertaamalla eri komponenttitasoja voidaan helposti nähdä esimerkiksi kahden komponentin välinen korrelaatio eli komponentit saavat samansuuruisia arvoja samoilla alueilla. Kuvassa 8 on esitetty esimerkki itseorganisoituvan kartan komponenttitasoista.



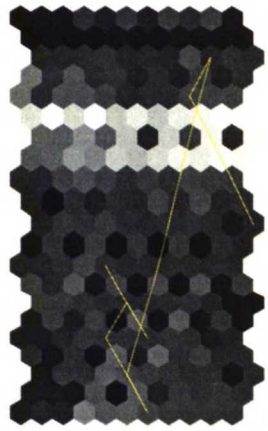

**Kuva 8.** SOM:n komponenttitasot.

Histogrammin avulla voidaan tarkastella syötedatan jakaumaa. Histogrammissa lasketaan jokaiselle karttayksikölle BMU:na esiintymisen määrä. Kuvassa 9 on esitetty kaksi itseorganisoituvan kartan histogrammia. Vasemmassa kuvassa histogrammi on piirretty SOM:n U-matriisin päälle. Suuret neliöt esittävät karttayksiköitä, joihin on sattunut eniten syötedatan näytteiden osumia. Pienet neliöt kuvaavat pieniä osumamääriä. Oikeanpuoleisessa kuvassa osumien määrä on esitetty pylväinä. Kuvasta nähdään, että histogrammit vastaavat toisiaan. Histogrammeja vertaamalla voidaan erilaisia datajoukkoja helposti verrata keskenään.



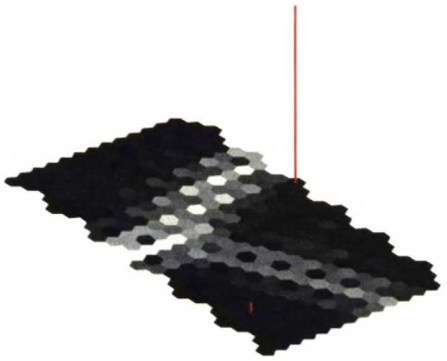
**Kuva 9.** SOM:n päälle piirretty histogrammi (vasemmalla) ja pylväshistogrammi (oikealla).

Aikariippuvissa sovelluksissa voidaan visualisoinnissa käyttää liikerata- eli trajektoriesitystä [11]. Trajektori on polku, jonka näytevektorien BMU:t muodostavat kartalle. Trajektoriesitystä voidaan käyttää esimerkiksi teollisuusprosessin valvonnassa. Itseorganisoituvan kartan klusterit kuvaavat tiettyjä prosessitiloja, jolloin trajektoreja piirtämällä voidaan helposti seurata prosessin siirtymisiä tilojen välillä. Kuvassa 10 on esimerkki trajektorin piirtämisestä.



**Kuva 10.** Trajektori esitettynä SOM:n U-matriisin päällä.

Mallivektoreille voidaan laskea kvantisointivirhe, kun halutaan tietää kuinka hyvin itseorganisoituva kartta kuvaa näytejoukkoa. Tällöin voidaan laskea keskimääräinen kvantisointivirhe eli etäisyys näytevektorin ja sen BMU:n välillä. Kuvassa 11 on yksi tapa esittää kvantisointivirhe.



**Kuva 11.** Kvantisointivirheen esittäminen.

## 3.4 Klusterointi

Kasaantumien löytäminen eli klusterointi on yksi itseorganisoituvan kartan tärkeimmistä sovelluksista [12]. SOM:n neuronit ovat jo itsessään klusterin keskipisteitä, mutta tulkinnan helpottamiseksi useista karttayksiköistä voidaan muodostaa suurempia klustereita. Merkittävä etu klusteroinnissa on se, että Voronoi-alueet ovat muodoltaan kuperia, kun taas usean karttayksikön muodostamat klusterit voivat olla myös ei-kuperia.

Yleinen menetelmä karttayksiköiden klusterointiin on laskea etäisyysmatriisi mallivektorien välillä ja käyttää matriisin suuria arvoja klusterin rajojen indikointiin. Tällaisen matriisin kolmiulotteisessa esitysmuodossa klusterit esiintyvät "laaksoina". Ongelmana on päättää, mitkä karttayksiköt muodostavat kunkin klusterin. Ratkaisuna on tyypillisesti käytetty kasaavia ja jakavia algoritmeja. Myös muita kuin etäisyyteen perustuvia menetelmiä voidaan käyttää.

Eräs mielenkiintoinen tapa klusteroida mallivektorit on käyttää toista itseorganisoituvaa karttaa. Tällaista rakennetta kutsutaan hierarkkiseksi SOM:ksi. Tavallisesti hierarkkisella SOM:lla tarkoitetaan karttapuuta, jonka alemmat tasot käyttäytyvät esikäsittelyasteina ylemmille tasoille. Hierarkkisen itseorganisoituvan kartan esitteli ensikerran Luttrell [13], joka osoitti, että vaikka lisäkerroksien lisääminen vektorikvantisaattoriin kasvattaa vääristymää rekonstruktiossa, se myös vähentää tehtävän monimutkaisuutta.

Itseorganisoituvaa karttaa voidaan käyttää luokittelussa määräämällä jokaiselle mallivektorille luokka ja päättämällä näytevektorin luokka sen BMU:n luokan mukaan. On kuitenkin muistettava, että luokittelu ei välttämättä onnistu, jos käytetään opetusdataa, jonka luokat jo tunnetaan. Tämä johtuu siitä, että itseorganisoituva kartta ei huomioi näytevektoreiden luokkia opetusvaiheessa.

### 3.5 Mallinnus

Perinteinen tapa lähestyä mallinnusta on estimoida piilotettua funktiota globaalisti. Viime vuosina on kuitenkin herännyt enemmän mielenkiintoa lokaaleja malleja kohtaan, koska monessa tapauksessa ne antavat parempia tuloksia kuin globaalit mallit. Tämä on tietysti luonnollista, jos estimoitavan funktion ominaisuudet vaihtelevat paljon piirreavaruuden eri osissa.

SOM:n syöteavaruudesta muodostama elastinen verkko voidaan ajatella olevan sokean etsimisen (blind lookup) malli, joka kuvaa näytedatan ilmiöitä. Mallia voidaan käyttää herkkyysanalyysiin. Mallin laajennuksessa paikallisia malleja sovitetaan kuhunkin karttayksikköön erikseen. Paikalliset mallit voidaan muodostaa monella eri tavalla. Mallien rakenne vaihtelee parhaasta esimerkkivektorista yksinkertaisiin ja pieniin monikerrosperseptroneihin. Tavallisesti paikalliset mallit ovat yksinkertaisia, kuten esimerkiksi lineaariset regressiomallit.

# 4 Pääkomponenttianalyysi

Pääkomponenttianalyysi (Principal Component Analysis, PCA) on perinteinen tilastollinen menetelmä, jota on paljon käytetty laajasti datan analysoinnissa ja kompressoinnissa. Alkuperäisestä, usein suurehkosta määrästä muuttujia muodostetaan lineaarikombinaatioita eli kertoimilla painotettuja summamuuttujia, joiden toivotaan selittävän mahdollisimman suuren osan alkuperäisten muuttujien vaihtelusta. Pääkomponentit valitaan siten, että ne ovat keskenään korreloimattomia. PCA on lineaarinen muunnosmenetelmä, jolla vähennetään datan ulottuvuutta. Seuraavassa on esitetty pääkomponenttimenetelmän periaatteet viitteen [5] mukaan.

Merkitään $m$-ulotteista satunnaisvektoria **X**:llä, joka esittää kiinnostavaa näytettä syöteavaruudesta. Lisäksi oletetaan, että satunaisvektori **X** on nollakeskiarvoinen eli

$$E[\mathbf{X}] = 0. \tag{7}$$

Jos **X** ei ole nollakeskiarvoinen, keskiarvo vähennetään satunnaismuuttujasta, jolloin saadaan nollakeskiarvoinen muuttuja. Seuraavaksi satunnaisvektori **X** projisoidaan yksikkövektorille **q** ($\|\mathbf{q}\| = 1$), mikä voidaan esittää sisätulona

$$A = \mathbf{X}^T\mathbf{q} = \mathbf{q}^T\mathbf{X}. \tag{8}$$

Projektio $A$ on satunnaismuuttuja, jonka keskiarvo ja varianssi ovat suhteellisia vektorin **X** statistiikkaan. Projektion $A$ keskiarvo on myös nolla ja varianssi

$$\sigma^2 = E\left[A^2\right] = \mathbf{q}^T\mathbf{R}\mathbf{q}, \tag{9}$$

jossa **R** on $m$ x $m$ –korrelaatiomatriisi (**R** = $E[\mathbf{X}\mathbf{X}^T]$). Pääkomponenttianalyysissä yritetään löytää lineaarikombinaatio $\mathbf{q}^T\mathbf{X}$, jonka varianssi on suurin. Tällöin saadaan ominaisarvo-ongelma

$$\mathbf{R}\mathbf{q} = \lambda\mathbf{q}, \tag{10}$$

jossa $\lambda$ on korrelaatiomatriisin ominaisarvo ja **q** on ominaisvektori. Yksinkertaisuuden vuoksi oletetaan, että vektorit $\lambda$ ovat selvästi eroavia. Nämä vektorit voidaan ratkaista esimerkiksi löytämällä ratkaisu yhtälöön

$$\det\left|\mathbf{R} - \lambda\mathbf{I}\right| = 0, \tag{11}$$

jossa **I** on identiteettimatriisi, jonka aste on sama kuin korrelaatiomatriisin **R**. Ominaisarvojen ja -vektoreiden ratkaiseminen on ei-triviaali tehtävä ja ongelman ratkaisemiseksi on olemassa useita menetelmiä. Yksi tapa on käyttää neuraalilaskentaa, kuten viitteessä [14] on tehty.

Järjestämällä ominaisarvot suuruusjärjestykseen voidaan muodostaa ortogonaalinen matriisi $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_m]$, joka toteuttaa ortonormaalisuusehdon

$$\mathbf{q}_i^T \mathbf{q}_j = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases}. \tag{12}$$

Ensimmäistä ominaisarvoa vastaava ominaisvektori osoittaa näin suunnan, jossa näytejoukolla on suurin varianssi ja edelleen suurin energia.

Merkitään satunnaisvektorin **X** realisaatiota **x**:llä. Yksikkövektorille **q** löytyy *m* kappaletta ratkaisuja, joten on mahdollista löytää *m* kappaletta projektiota. Yhtälön (8) perusteella voidaan kirjoittaa

$$a_j = \mathbf{q}_j^T \mathbf{x} = \mathbf{x}^T \mathbf{q}_j \qquad j = 1, 2, \ldots, \mathrm{m}, \tag{13}$$

missä $a_j$ on **x**:n projektio pääsuuntaan, jonka yksikkövektori $\mathbf{q}_j$ määrittää. Muuttujia $a_j$ kutsutaan pääkomponenteiksi. Ensimmäinen pääkomponentti on suurinta ominaisarvoa vastaavan ominaisvektorin projektio.

Alkuperäisen datan rekonstruointi projektioista tapahtuu yhdistämällä projektiot samaan vektoriin

$$\begin{aligned} a &= [a_1, a_2, \ldots, a_m]^T \\ &= [\mathbf{x}^T \mathbf{q}_1, \mathbf{x}^T \mathbf{q}_2, \ldots, \mathbf{x}^T \mathbf{q}_m]^T \\ &= \mathbf{Q}^T \mathbf{x} \end{aligned} \tag{14}$$

ja ratkaisemalla **x**, kun tiedetään, että $\mathbf{Q}^T = \mathbf{Q}^{-1}$. Tulokseksi saadaan

$$\mathbf{x} = \mathbf{Q}\mathbf{a} = \sum_{j=1}^{m} a_j \mathbf{q}_j . \tag{15}$$

Sen sijaan, että käytetään kaikkia kovarianssimatriisin ominaisvektoreita, voidaan data esittää muutaman ortogonaalisen kantavektorin avulla.

Alkuperäisen datan ulotteisuutta voidaan vähentää laskemalla alkuperäiselle datalle **x** pääkomponenttien sarja seuraavasti:

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_l \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_l^T \end{bmatrix} \mathbf{x}, \qquad l \leq m, \tag{16}$$

Yhtälössä käytetään $l$ suurinta ominaisvektoria, jolloin saadaan $l$ kappaletta pääkomponentteja. Tämä tarkoittaa, että alkuperäinen data projisoidaan $l$-ulotteiseen koordinaattiakselistoon. Tällä tavoin minimoidaan keskimääräinen neliöllinen virhe datan ja annetulla määrällä ominaisvektoreita muodostetun projektion välillä.

# 5 Lähimmän naapurin menetelmä

Lähimmän naapurin menetelmä [3, 6] (Nearest Neighbor method, kNN) on hyvin suoraviivainen ja yksinkertainen luokittelumenetelmä: valitaan $k$ lähintä opetusjoukon pistettä ja määrätään **x**:n luokka näiden $k$ datapisteen enemmistön mukaan. Lähin etäisyys tarkoittaa tässä tapauksessa $p$-ulotteisessa syöteavaruudessa laskettua etäisyysmittaa. Toisin sanoen opetusjoukosta etsitään niitä kohtia, jotka ovat mahdollisimman samankaltaisia kuin syötemuuttujat ja sen jälkeen uusi näyte luokitellaan voimakkaimmin esiintyvän luokan mukaan. Algoritmi määritellään seuraavasti:

1. Identifioidaan tuntemattoman syötevektorin **x** $k$ lähintä naapuria N opetusvektorin joukosta. $k$ valitaan yleensä siten, että se ei ole luokkien määrän $M$ monikerta.
2. Lasketaan vektoreiden $k_i$ määrä , jotka kuuluvat luokkaan $\omega_i$, $i$ = 1, 2, ..., $M$. naapurista. Näin $\sum_i k_i = k$ .
3. Määritetään x:n kuuluvan luokaan $\omega_i$, jolla on suurin arvo $k_i$.

Teoreettisesta näkökulmasta tarkasteltuna voidaan ajatella käsiteltävän muuttujaavaruuden pientä tilaa, joka on keskitetty syötevektoriin **x** ja jonka säde on etäisyys $k$:nteen lähimpään naapuriin. Tämän pienen tilan eri luokkien todennäköisyyden maximun likelihood -esimaattorit saadaan tilassa sijaitsevien opetusjoukon pisteiden eri luokkien osuutena. Lähimmän naapurin menetelmä luokittelee uuden näytevektorin siihen luokkaan jolla on suurin estimoitu todennäköisyys.

Vaikka menetelmä näyttää erittäin yksinkertaiselta, jättää se paljon valintaa. Käytännössä käytettävä etäisyysmitta ja lähimpien naapurien määrä $k$ täytyy valita. Etäisyysmittana useimmat sovellukset käyttävät kaavan (2) euklidista etäisyyttä. Yksinkertaisin luokitin voidaan toteuttaa, kun valitaan $k = 1$. Tällöin saadaan kuitenkin epästabiili luokitin, jolla on suuri varianssi ja se on herkkä datalle. Kuitenkin, jos opetusjoukon koko on tarpeeksi suuri, luokitin toimii hyvin. Kasvattamalla $k$:n arvoa voidaan vähentää varianssia, mutta toisaalta harha voi kasvaa. Naapureiden määrän $k$ valinnasta on olemassa teorioita, mutta käytännössä valinta riippuu datan rakenteesta, ja kokeileminen näyttää olevan paras keino.

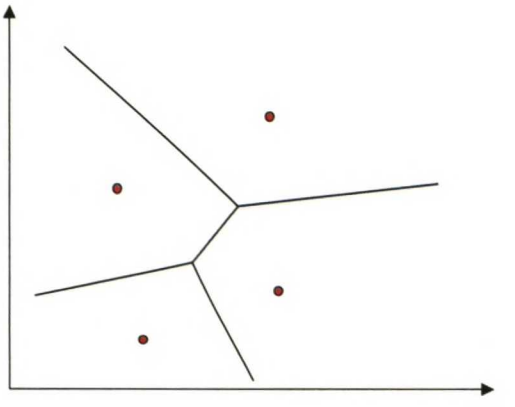
Lähimmän naapurin menetelmällä on useita hyviä ominaisuuksia. Se on helppo ohjelmoida ja algoritmin optimointia tai opetusta ei tarvita. Joissakin ongelmissa luokittelutarkkuus voi olla hyvä verrattuna monimutkaisempiin menetelmiin kuten neuroverkot. Useiden luokkien tapauksissa lähimmän naapurin menetelmä on edelleen käyttökelpoinen. Puuttuvat näytteet eivät vaikuta luokittimen toimintaan; luokitin toimii siinä aliavaruudessa, joka sisältää vain saatavilla olevan datan.

Suuriulottuvuuksinen data aiheuttaa ongelmia. Suurella määrällä muuttujia lähimmät $k$ pistettä voivat olla todellisuudessa melko kaukana. Toinen heikkous on siinä, että luokitin ei muodosta mallia datasta. Jos opetusjoukko on suuri, lähimpien naapurien etsiminen joukosta voi olla paljon aikaa kuluttava prosessi. Menetelmä vaatii myös paljon muistia suuren opetusjoukon tallentamiseen.

# 6 Oppiva vektorikvantisointi

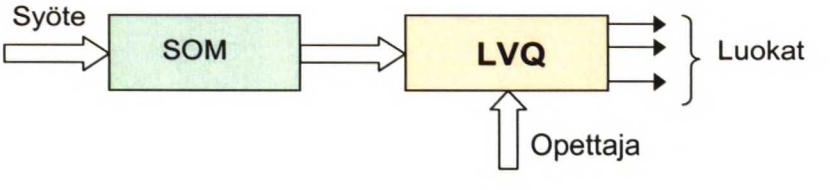
Oppiva vektorikvantisointi (Learning Vector Quantization, LVQ) [2, 5] on tekniikka, joka hyödyntää syötevektorien piilotetun rakenteen datan pakkausta varten. Itse asiassa, syöteavaruus jaetaan erilaisiin alueisiin ja jokaiselle alueelle määritellään rekonstruointivektori. Kun kvantisaattorille annetaan syötevektori, aluksi määritetään missä alueessa se sijaitsee ja sen jälkeen palautetaan alueen esitysvektori. Käyttämällä tätä esitysvektoria syötevektorin sijasta voidaan dataa kompressoida ja siten säästetään tallennustilaa tai kaistanleveyttä. Esitysvektoreiden joukkoa kutsutaan koodikirjaksi (code book) ja yhtä esitysvektoria koodisanaksi (code word).

Vektorikvantisaattoria, jolla on pienin koodauspoikkeama, kutsutaan Voronoi- tai lähimmän naapurin kvantisaattoriksi. Tällöin syöteavaruuden pistejoukoista muodostuneet Voronoi-solut vastaavat sitä avaruuden osaa, joka saadaan euklidiseen metriikkaan perustuvalla lähimmän naapurin säännöllä (kts. edellinen kappale). Kuvassa 12 nähdään esimerkki syöteavaruudesta, joka on jaettu neljään Voronoi-soluun. Jokaiseen soluun on piirretty Voronoi-vektori (esitysvektori). Jokainen solu sisältää ne syöteavaruuden pisteet, jotka ovat lähimpänä Voronoi-vektoria.



**Kuva 12.** Voronoi-solut ja niiden esitysvektorit.

SOM-algoritmilla voidaan ohjaamattomasti laskea likimääräiset Voronoi-vektorit, jotka vastaavat SOM:n synaptisia painovektoreita. Piirrekartan laskemisen voidaan kuvitella olevan ensimmäinen vaihe kaksivaiheisesta hahmontunnistusongelman ratkaisusta. Toisessa vaiheessa LVQ:n avulla hienosäädetään itseorganisoituvaa piirrekarttaa ja siten parannetaan luokittimen tarkkuutta. Kuvassa 13 on esitetty luokittimen lohkokaavio.




**Kuva 13.** Adaptiivisen hahmontunnistimen lohkokaavio.

Oppiva vektorikvantisointi on ohjatun oppimisen tekniikka, joka hyödyntää luokkainformaatiota Voronoi-vektoreiden säätämisessä. Tällä tavalla parannetaan luokittimen päätösalueiden laatua. Syötevektori **x** valitaan satunnaisesti syöteavaruudesta. Jos syötevektorin luokka on sama kuin Voronoi-vektorin **w**, Voronoi-vektoria siirretään kohti

syötevektoria **x**. Toisaalta, jos luokka on eri, Voronoi-vektoria **w** siirretään näytevektorista **x** poispäin.

Merkitään Voronoi-vektoreiden joukkoa $\{\mathbf{w}_j\}_{j=1}^{l}$:llä ja syötevektoreiden joukkoa $\{\mathbf{x}_i\}^{N}{}_{i=1}$:llä. Lisäksi oletetaan, että syötevektoreita on paljon enemmän kuin Voronoi-vektoreita, mikä on tyypillistä käytännössä. LVQ-algoritmi toimii seuraavasti:

1. Oletetaan, että Voronoi-vektori $\mathbf{w}_c$ on lähinnä näytevektoria $\mathbf{x}_i$. Merkitään vektoreiden luokkia symboleilla $\omega_{wc}$ ja $\omega_{xi}$. Voronoi-vektoria $\mathbf{w}_c$ säädetään seuraavasti:

   Jos $\omega_{wc} = \omega_{xi}$, silloin

   $$\mathbf{w}_c(n+1) = \mathbf{w}_c(n) + \alpha_c(n)[\mathbf{x}_i(n) - \mathbf{w}_c(n)] \tag{17}$$

   Jos toisaalta $\omega_{wc} \neq \omega_{xi}$, silloin

   $$\mathbf{w}_c(n+1) = \mathbf{w}_c(n) - \alpha_c(n)[\mathbf{x}_i(n) - \mathbf{w}_c(n)] \tag{18}$$

2. Muita Voronoi-vektoreita ei modifioida.

Ongelma on siinä, kuinka määritetään $\alpha_c$ siten, että konvergoituminen on nopeinta. Kaavat (17) ja (18) voidaan yhdistää muotoon

$$\mathbf{w}_c(n+1) = [1 - s(n)\alpha_c(n)]\mathbf{w}_c(n) + s(n)\alpha_c(n)x_i(n)], \tag{19}$$

missä $s(n) = +1$, kun luokittelu on oikein ja $s(n) = -1$, kun luokittelu on väärin. On selvää, että koodikirjavektoreiden tilastollinen tarkkuus on lähes optimaalinen silloin, kun kaikilla näytteillä on sama paino. Toisin sanoen, jos opetusjakson aikana eri ajan hetkillä tehdyillä korjauksilla on sama voimakkuus. Opetusaskeleessa $\mathbf{x}(n)$:n viimeisen jäljen voimakkuus skaalataan kertoimella $\alpha_c$ ja esimerkiksi saman askeleen aikana $\mathbf{x}(n-1)$:n jälki skaalataan termillä $[1 - s(n)\,\alpha_c(n)]\,\alpha_c(n-1)$. Nyt voidaan merkitä nämä kaksi skaalausta identtisiksi:

$$\alpha_c(n) = [1 - s(n)\alpha_c(n)]\alpha_c(n-1) \tag{20}$$

Jos tämä pätee kaikilla $n$:n arvoilla, voidaan osoittaa että $\mathbf{x}(n)$:n jäljet, jotka on tallennettu kierrokseen $n$ mennessä, ovat skaalautuneet samansuuruisesti opetusvaiheen lopussa. Optimiarvot määritellään rekursiivisesti

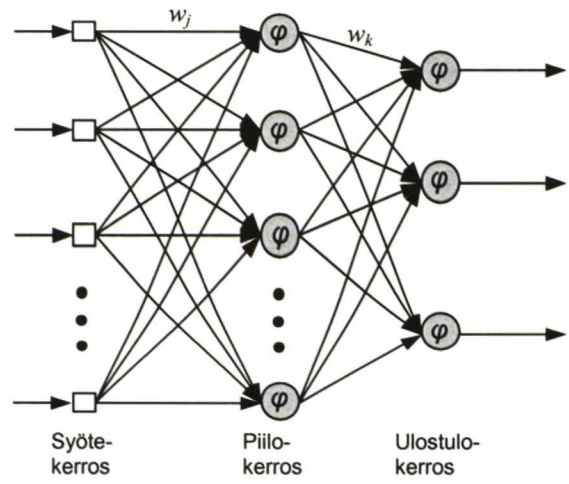
$$\alpha_c(n) = \frac{\alpha_c(n-1)}{1 - s(n)\alpha_c(n-1)}. \tag{21}$$

Koska $\alpha_c$ voi myös kasvaa on oleellista, ettei se kasva ykköstä suuremmaksi. Tämä tilanne voidaan nähdä algoritmista itsestään. Alkuarvoksi voidaan valita 0,5, mutta vähintään yhtä hyvä arvaus on 0,3.

# 7 Monikerrosperseptroni-neuroverkko

Eteenpäin syöttävät monikerrosperseptronit (Feedforward Multilayer Perceptrons, MLP) [3, 5] ovat laajimmin käytettyjä malleja keinotekoisten neuroverkkojen joukossa. MLP-rakenne tuottaa epälineaarisen kuvauksen reaaliarvoisesta syötevektorista **x** reaaliarvoiseen vastevektoriin **y**. Tuloksena MLP-verkkoa voidaan käyttää ei-lineaarisena mallina regressio-ongelmiin, kuten luokitteluun.

MLP-verkko koostuu sensoriyksiköistä, jotka muodostavat sisäänmenokerroksen, yhden tai useamman piilokerroksen ja ulostulokerroksen. Sensoriyksiköt eli neuronit muodostavat syötteistä painotetun summan, joka muunnetaan epälineaarisesti. Syötesignaali etenee verkossa eteenpäin kerros kerrokselta ja siitä nimitys eteenpäin syöttävä (feedforward). Kuvassa 14 on esimerkki monikerrosperseptroni-neuroverkosta, jossa on yksi piilokerros. Kuvassa oleva verkko on täysin kytketty eli jokaisesta edellisen kerroksen neuronista on yhteys kaikkiin seuraavan kerroksen neuroneihin.




**Kuva 14.** Esimerkki MLP-neuroverkosta.

Matemaattisesti syötemuuttujien $x_j$ ja vasteen $y$ välinen yhteys voidaan kirjoittaa

$$y = \varphi_l\left(\sum_k w_k^{(2)} \varphi_k\left(\sum_j w_j^{(1)} x_j\right)\right), \tag{22}$$

missä $w_k$ ja $w_j$ ovat eri kerrosten painoja ja $\phi_l$ ja $\phi_k$ ovat neuroneiden epälineaarisia aktivaatiofunktioita. Epälineaarisuus on olennaista, koska muuten malli supistuu lineaarikombinaatioiden verkottuneeksi ketjuksi, mikä on yksinkertaisesti lineaarinen kombinaatio. MLP-verkoissa käytetään "pehmeää" epälineaarisuutta eli aktivaatiofunktio on differentioituva kaikkialla. Yleisesti käytetty epälineaarinen funktio, joka täyttää vaatimuksen, on sigmoid-epälineaarisuus, joka on määritelty logaritmisena funktiona:

$$y = \frac{1}{1+\exp(-v_j)}, \tag{23}$$

missä $v_j$ on painotettu summa kaikista synaptisista syötteistä ja harhasta.

Neuroverkon kerrosten määrällä ei ole rajoituksia, mutta on ositettu, että yhden piilokerroksen verkolla voidaan mallintaa mitä tahansa jatkuvaa funktioita. Käytännössä kerrosten määrä riippuu käytettävästä datasta ja muista syistä kuten tulkittavuudesta. On olemassa myös verkkoja, joissa kerrosten välinen yhteys ei ole pelkästään peräkkäisten kerrosten välillä vaan kytkentöjä voi olla aikaisemmista kerroksista.[3, 5]

Monikerrosperseptroni-verkkoja on sovellettu moniin vaikeisiin ongelmiin opettamalla verkkoa ohjatun oppimisen menetelmällä, joka perustuu laajasti tunnettuun virheen takaisinvirtaus (error back-propagation) –algoritmiin. Algoritmi perustuu virhe-korjaus -oppimissääntöön (error-correction learning rule). Takaisinvirtausoppiminen sisältää kaksi etenemisvaihetta verkon kerrosten läpi: eteenpäin siirtyminen ja taaksepäin siirtyminen. Eteenpäin siirtymisessä syöte kulkee verkon kerrosten läpi ja tuottaa ulostulossa vasteen. Etenemisen aikana verkon synaptiset painovektorit ovat vakiot. Taaksepäin etenemisvaiheessa synaptisia painovektoreita säädetään virhe-korjaus –säännön mukaan. Virhesignaali saadaan vähentämällä ulostulossa saatu vaste halutusta vasteesta. Tämä virhesignaali kuljetetaan sitten taaksepäin verkon läpi synaptisten kytkentöjen suunnan vastaisesti – tästä nimitys virheen takaisinvirtaus. Synaptisia painoja muutetaan siten, että neuroverkon vaste siirtyy lähemmäs haluttua vastetta tilastollisessa mielessä.

Virheen takaisinvirtaus –algoritmi voidaan esittää opetusnäytteelle $\{(\mathbf{x}(n),\mathbf{d}(n))\}_{n=1}^{N}$ seuraavan proseduurin mukaisesti:

1. *Alustus*. Synaptiset painot ja kynnysarvot poimitaan tasajakaumasta, jonka keskiarvo on nolla. Jakauman varianssi valitaan siten, että neuronien muodostamien paikalliskenttien (kaikkien synaptisten syötteiden painotettu summa lisättynä harhalla) keskihajonta sijaitsee sigmoid-aktivaatiofunktion lineaarisen ja kyllästyneen alueen välissä.
2. *Opetusnäytteen syöttäminen*. Verkolle syötetään opetusnäyte. Jokaiselle näytteelle suoritetaan kohtien 3 ja 4 eteenpäin ja taaksepäin laskennat.
3. *Eteenpäin laskenta*. Lasketaan verkon muodostamat paikalliset kentät ja funktiosignaalit etenemällä verkon läpi kerros kerroksella. Indusoitunut paikalliskenttä neuronille $j$ kerroksessa $l$ on

$$v_j^{(l)}(n) = \sum_{i=0}^{m_0} w_{ji}^{(l)}(n) y_i^{(l-1)}(n), \tag{24}$$

missä $y_i^{(l-1)}(n)$ on neuronin $i$ ulostulosignaali edellisellä kerroksella $l - 1$ ja $w_{ji}^{(l)}(n)$ on neuronin $j$ synaptinen paino kerroksessa $l$ syötettynä kerroksen $l - 1$ neuronista $i$. Kun i = 0, $y_0^{(l-1)}(n) = +1$ ja $w_{j0}^{(l)}(n) = b_j^{(l)}(n)$ on harha (eng. bias) kytkettynä neuroniin $j$ kerroksessa $l$. Ulostulo signaali on

$$y_j^{(l)} = \varphi_j(v_j(n)), \tag{25}$$

kun käytetään sigmoid-aktivaatiofunktiota. Jos neuroni $j$ on ensimmäisessä piilokerroksessa

$$y_j^{(0)} = x_j(n), \tag{26}$$

missä $x_j(n)$ on syötevektorin $\mathbf{x}(n)$ elementti. Lasketaan virhesignaali halutun signaalin ja ulostulon (kerros $L$) erotuksena

$$e_j(n) = d_j(n) - y_j^{(L)}. \tag{27}$$

4. *Taaksepäin laskenta.* Lasketaan verkon paikalliset gradientit $\delta$, jotka määritellään

$$\delta_j^{(l)} = \begin{cases} e_j^{(L)}(n)\varphi_j'(v_j^{(L)}(n)) \\ \varphi_j'(v_j^{(l)}(n))\sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \end{cases}, \tag{28}$$

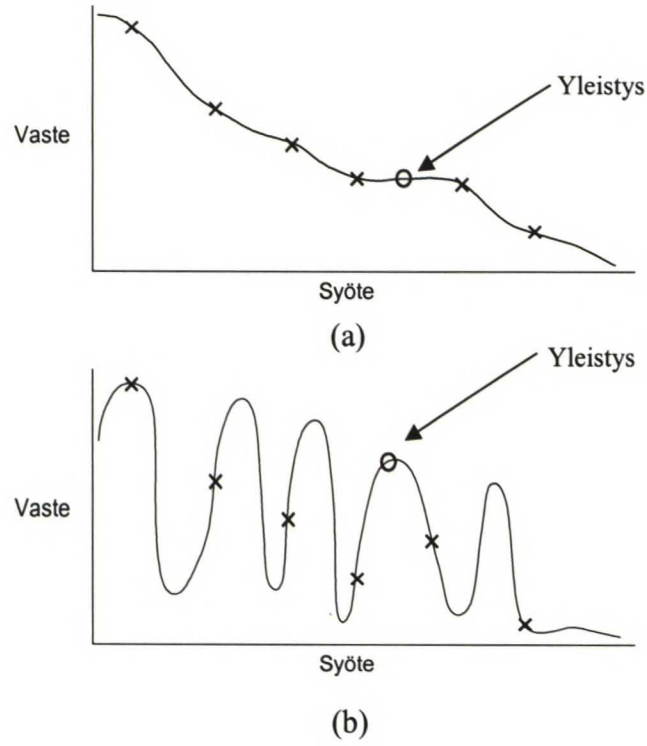
missä ylempi yhtälö pätee ulostulokerrokselle ja alempi piilokerroksille. Heittomerkki tarkoittaa derivaattaa suluissa olevan argumentin suhteen. Lopuksi säädetään kerroksen l synaptisia painoja yleisen deltasäännön mukaan

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha[w_{ji}^{(l)}(n-1)] + \eta\delta_j^{(l)}(n) y_i^{(l-1)}(n),$$

missä $\eta$ on opetuskerroin ja $\alpha$ on impulssivakio.

Opetusnäytteiden syöttäminen pitää tehdä satunnaisessa järjestyksessä epookki kerrallaan. Impulssivakiota ja opetuskerrointa säädetään iteraatiokierrosten aikana.

Monikerrosperseptronilla halutaan olevan hyvä yleistettävyys, mikä tarkoittaa sitä, että verkon laskema syöte-vaste –yhteys on oikea testidatalle, jota ei ole käytetty ollenkaan verkon rakentamiseen tai opettamiseen. Verkko voi oppia liikaa opetusdatasta, jolloin verkkoon syntyy muisti opetusdatasta. Malli kuvaa tällöin hyvin opetusdataa, mutta yleistettävyys kärsii, kuten kuvasta 15 voidaan nähdä. Tapahtumaa kutsutaan ylioppimiseksi.

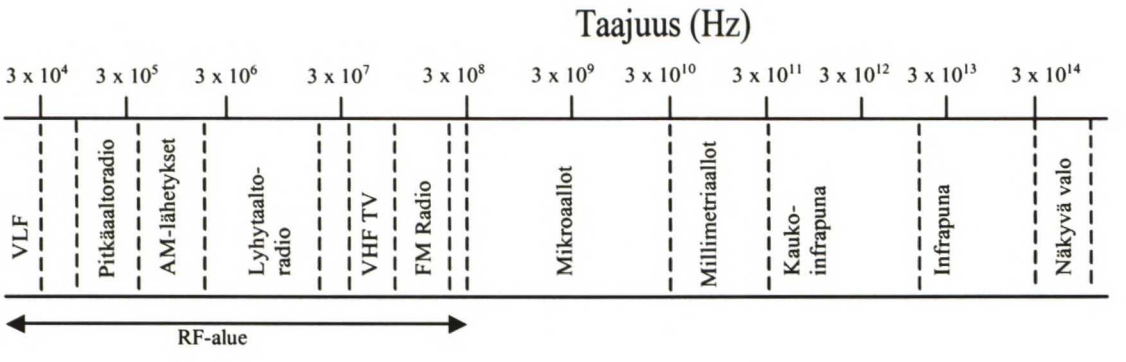



**Kuva 15.** (a) Hyvin yleistävä malli ja (b) ylioppinut malli (huono yleistävyys).

# 8 RF- ja mikroaaltospektristä kerätty data

## 8.1 RF- ja mikroaallot

Kuvassa 16 nähdään RF- ja mikroaaltoalueiden sijoittuminen sähkömagneettisessa spektrissä. Radioaallot kattavat taajuusalueen 3 kHz – 300 GHz. Lyhenne RF tulee sanoista radio frequency ja se tarkoittaa radioaaltoja, jotka ovat taajuusalueella 3 kHz – 300 MHz. Mikroaaltojen taajuusalue on 300 MHz – 30 GHz. Taulukossa 1 on esitetty radioaaltojen taajuusalueet.[15]




**Kuva 16.** Sähkömagneettinen spektri.

Nykyaikainen radiotietoliikenne on keskittynyt erityisesti UHF- ja SHF-alueille, mutta myös mikroaaltoalueen käyttö on yleistymässä. Myös muut radiosovellukset kuten tutkat, anturit ja tehosovellukset ovat keskittyneet VHF-, UHF- ja SHF-alueille. Muiden alueiden merkitys on vähenemässä.

Tietoliikenne on radiotekniikan tärkein hyödyntämiskohde. Yleisradiotoiminnassa käytetään VHF- ja UHF-alueita. Satelliittilähetykset toimivat mikroaaltoalueella 12 GHz:n kaistalla. Voimakkaimmin kasvanut radiotietoliikenteen alue on kuitenkin matkaviestintä. Toisen sukupolven matkaviestinjärjestelmät käyttävät taajuuksia aina 2 GHz:iin asti.

**Taulukko 1.** Radioaallot.

| | | |
|---|---|---|
| VLF | Very Low Frequencies | 3 – 30 kHz |
| LF | Low Frequencies | 30 – 300 kHz |
| MF | Medium Frequencies | 300 – 3000 kHz |
| HF | High Frequencies | 3 – 30 MHz |
| VHF | Very High Frequencies | 30 – 300 MHz |
| UHF | Ultra High Frequencies | 300 – 3000MHz |
| SHF | Super High Frequencies | 3 – 30 GHz |
| EHF | Extreme High Frequencies | 30 – 300 GHz |

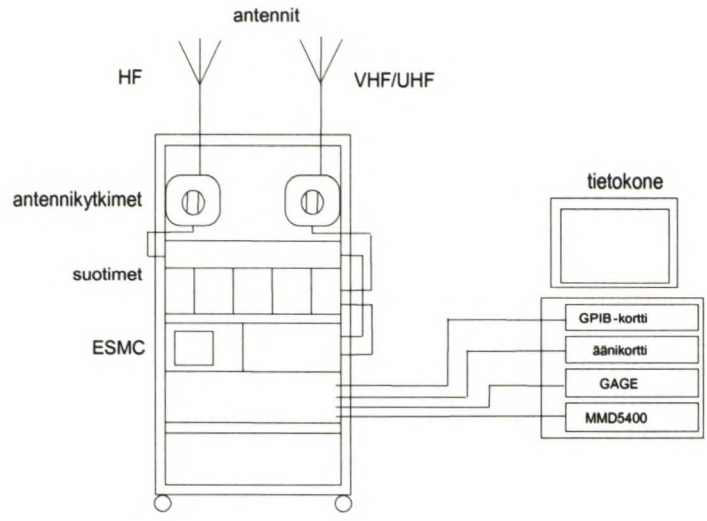
Maailmanlaajuisesti toimivat satelliittipuhelinjärjestelmät toimivat 2 GHz:n taajuudella. Langattomien lähiverkkojen (WLAN = wireless local area network) merkitys on kasvanut kiinteiden lähiverkkojen korvaajana. Tärkein taajuuskaista on 2,4 GHz. Muita lähiverkkojen taajuusalueita ovat 5 ja 17 GHz sekä 61 GHz.

Kiinteitä radiolinkkejä käytetään yleisissä ja Puolustusvoimien televerkoissa. Radiolinkit ovat nykyään yleensä digitaalisia ja ne toimivat mikroaaltoalueella. Tärkeimpiä taajuusalueita ovat 13, 15, 18, 23, 26, 38, 55 ja 58 GHz. Suurimmat taajuudet soveltuvat lyhyille matkoille. Kiinteissä satelliittilinkeissä käytetään 6/4, 14/11 ja 30/20 GHz:n kaistoja.

Tutkalla on useita sotilas- ja siviilisovelluksia. Perinteisesti tutkat ovat toimineet mikroaaltoalueella, mutta millimetriaaltoalueen (30 – 300 GHz) käyttö on kasvanut. Maailmanlaajuisia radionavigointijärjestelmiä ovat GPS (Global Positioning System) ja GLONASS (Global Navigation Satellite System), jotka käyttävät taajuuksia 1,2 ja 1,6 GHz.

## 8.2 Signaalinkeruujärjestelmä

Työssä käytetty testiaineisto on tallennettu PC –pohjaisella signaalinkeruujärjestelmällä. Järjestelmä koostuu antennista, vastaanotinjärjestelmästä, signaalin tallennuslaitteistosta ja ohjausosasta. Järjestelmä on koottu kaupallisista tuotteista. Kuvassa 17 on esitetty signaalinkeruujärjestelmän rakenne.




**Kuva 17.** Signaalinkeruujärjestelmä.

Antennijärjestelmä koostuu sekä VHF/UHF- että HF-alueen antenneista. VHF/UHF-alueen antenni on log-periodinen ja sen taajuusalue on 25 MHz – 3 GHZ. Antenni on suuntaava ja sen polaarisuutta voidaan vaihtaa. HF-antenni on invertoitu V-

dipoliantenni. Taajuusalue HF-antennilla on 1,5 – 30 MHz. Antennikaapelit ovat koaksiaalia. Antennit on kytketty antennikytkimien ja suodattimien kautta vastaanottimeen. Suodattimien tehtävänä on poistaa häiriöitä ja siten parantaa signaali-kohinasuhdetta.

Vastaanotinjärjestelmän ydin on Rohde & Schwarz ESMC-supervastaanotin, jonka taajuusalue ESMC-FE –laajennusosalla on 0,5 MHz – 3,0 GHz. Lisäksi vastaanotin järjestelmään kuuluu RF-suodatinmatriisi. Vastaanotin kykenee demoduloimaan SSB-, CW-, AM-, FM-, LOG- ja pulssimoduloituja signaaleja. Vastaanotettava signaalitaso voi olla väliltä –120...0 dBm. Hakunopeus järjestelmällä on 16000 kanavaa sekunnissa reaaliaikaisesti ja offline-jatkokäsittelyyn saadaan tallennettua 8 MHz:n kaista. Vastaanotin, suotimet ja antennikytkimet on sijoitettu laitekehikkoon

Signaalin tallennus tapahtuu tietokoneen mittauskortilla. Mittauskortti on PCI-väyläöinen GAGE Compuscope 12100 –kortti 8 MB:n muistilla. Tallennus tehdään 12 bittisesti ja kortissa on kaksi kanavaa. Maksimi kaistanleveys RF-signaalilla on 8 MHz ja 40 MHz kantataajuudella. Tallennusjärjestelmä pystyy tallentamaan maksimissaan 120 s kestoisen signaalin kaistanleveyden ollessa alle 2,5 MHz johtuen järjestelmän muistin määrästä. Käytettäessä leveämpää kaistaa tallennusaika pienenee huomattavasti. Tiedonkeruujärjestelmää ohjataan tietokoneen ohjelmistoilla. Vastaanottimelle, antennin kääntömoottorille ja signaalintallennuskortille on omat ohjelmansa.

## 8.3 Signaalidatan formaatti ja ominaisuudet

Signaalin tallennusjärjestelmä tallentaa kerättävän signaalin GageScope Signal File Format (.SIG) –muotoon. Tiedosto on binääritiedosto, joka sisältää 512 tavun mittaisen otsikko-osan ja mielivaltaisen määrän datapisteitä. Datapisteet ovat vastaanottimen välitaajuussignaalin jännitetasoja näytteenottohetkellä, joten näyte on yksiulotteinen aikasarja. Välitaajuussignaalin taajuus on 1,4 MHz ja näytteenottotaajuus on 5,0 MHz. Tässä työssä tutkittavien RF-signaalien pituus on 4194176 näytepistettä. Laskennan keventämiseksi näytteistä valittiin tutkittavaksi 100000 ensimmäistä arvoa. Datan analysoimiseksi SIG-tiedostot käännetään Matlab-ohjelmiston ymmärtämään ASCII-formaattiin.

# 9 Piirrevektorin valinta

Piirreirrotus on luokittelun ja hahmontunnistuksen kannalta oleellinen prosessi. Kirjallisuudessa on esitetty useita erilaisia lähinnä viestisignaaleista laskettuja piirteitä, joita käytetään signaalin modulaatiolajin tunnistamiseen ja luokitteluun. Lähetelajin tai signaalin sisällön (musiikki, puhe, sähkötys, data jne.) tunnistamiseksi kehitettyjä piirteitä löytyy huomattavasti vähemmän. Piirteitä ja piirrevalintaa on käsitelty viitteissä [3] ja [6].

Piirteitä voivat olla luokitteluun tai visualisointiin käytetyt mittaukset tai piirteet voidaan muodostaa mittauksia prosessoimalla. Yleensä piirteet $x_n$ esitetään piirrevektorina

$$\mathbf{x} = [x_1, x_2, ..., x_n]^T, \tag{29}$$

missä $T$ tarkoittaa transpoosia. Jokainen piirrevektori määrittelee yksiselitteisesti yhden hahmon (muodon). Piirteet ja niistä muodostetut piirrevektorit voidaan ajatella olevan satunnaismuuttujia, koska luonnollisesti havainnot eri hahmoista aiheuttavat satunnaisvaihtelua. Satunnaisvaihtelu johtuu osittain havainnointivälineen mittauskohinasta ja osittain kunkin hahmon erilaisista ominaisuuksista.

Piirteitä voidaan generoida näytteistä esimerkiksi lineaaristen muunnosten avulla. Peruskonsepti on muuttaa annettu näytejoukko uudeksi piirrejoukoksi. Mikäli muunnos on valittu sopivasti, muunnospohjaisten piirrevektoreiden sisältämä informaatio on tiheästi pakattua verrattuna alkuperäisiin näytteisiin. Tämä tarkoittaa, että luokittelussa ja visualisoinnissa olennainen tieto on puristettu suhteelliseen pieneen määrään piirteitä. Tällä tavoin myös piirreavaruuden dimensio pienenee.

Piirteen muodostamista tarkastellaan kahdesta eri lähtökohdasta. Ensimmäinen lähestymistapa on käyttää aika- tai taajuustason esitystä piirteiden generointiin. Tätä lähestymistapaa on käytetty yleensä modulaation tunnistamiseen. Toisessa menettelytavassa piirteet muodostetaan aallokemuunnoksen (wavelet transform) ja sen sovelluksen avulla.

## 9.1 Modulaation tunnistamisessa käytettyjä piirteitä

### 9.1.1 Verhokäyrän varianssin ja neliöllisen keskiarvon suhde

Piirteiden irrottaminen aikariippuvaisista signaaleista perustuu usein aika- ja taajuusesitysten hyödyntämiseen. Piirteiden muodostus tehdään analyyttisen verhokäyrän avulla ja siksi se vaatii Hilbert-muunnoksen [16] laskemista.

Hilbert-muunnoksen avulla reaalisesta datajoukosta voidaan muodostaa analyyttinen signaali $x = x_r + jx_i$, missä $x_r$ on alkuperäinen data ja $x_i$ imaginääri osa, joka saatu Hilbertin-muunnoksella. Signaalin $x(t)$ Hilbert-muunnos $\hat{x}(t)$ määritellään yhtälöllä

$$\hat{x}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(s)}{t-s} ds , \tag{30}$$

missä integraali on Cauchyn pääarvointegraali. Hilbert-muunnos tuottaa –90 asteen vaihesiirron signaalin $x(t)$ positiivisille taajuuksille ja +90 asteen vaihesiirron negatiivisille taajuuksille. Kaikkien taajuuskomponenttien amplitudit säilyvät muuttumattomina. Hilbrt-muunnoksella $\hat{x}(t)$ on sama amplitudispektri ja autokorrelaatiofunktio kuin signaalilla $x(t)$ ja lisäksi $x(t)$ ja $\hat{x}(t)$ ovat keskenään ortogonaalisia.

Hilbert-muunnos on hyödyllinen työkalu, kun lasketaan hetkittäisiä arvoja aikasarjoista, erityisesti amplitudia ja taajuutta. Hetkellinen amplitudi on kompleksisen Hilbert-muunnoksen amplitudi ja hetkellinen taajuus on hetkittäisen vaihekulman muutosnopeus. Esimerkiksi puhtaalle siniaallolle amplitudi ja taajuus ovat vakioita.

Hilbert-muunnoksen avulla voidaan laskea esimerkiksi seuraavat piirteet: hetkellisen amplitudin keskiarvo $\overline{A}_s$ ja varianssi $\sigma_A^2$ sekä hetkellisen taajuuden keskiarvo $\bar{f}_s$ ja varianssi $\sigma_f^2$. Varianssi ja keskiarvo lasketaan käytännössä keskiarvojen avulla kaavoista

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{31}$$

$$\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \,. \tag{32}$$

Viitteessä [17] on esitetty analogisen modulaatiolajin (AM, FM, DSB ja SSB) tunnistukseen käytetty piirremuuttuja $R$, joka voidaan kirjoittaa

$$R = \frac{\mathrm{var}\{V^2\}}{(E\{V^2\})^2}, \tag{33}$$

missä $V$ on signaalin $x(t)$ analyyttinen verhokäyrä. Verhokäyrä saadaan kaavasta

$$V^2 = x^2 + \hat{x}^2, \tag{34}$$

missä $\hat{x}$ on Hilbert-muunnos signaalista $x$. Edelleen viitteessä [18] on menestyksekkäästi käytetty neljää erilaista piirremuuttujaa modulaatiolajien tunnistukseen. Piirremuuttujat pohjautuvat kaavoihin (33) ja (34).

Parametri $R_2$ on lähes sama kuin edellä oleva parametri $R$, mutta $R_2$:n laskemiseen käytetään parametri-invarianttista suodatinta $Q$ Hilbert-muunnoksen sijasta. $R_2$ voidaan kirjoittaa

$$R_2 = \frac{\mathrm{var}\{Q[x]\}}{(E\{Q[x]\})^2}, \tag{35}$$

missä Q[x] on muotoa

$$Q[x] = \left| x_{n-1}^2 - x_n x_{n-2} \right|. \tag{36}$$

Signaalin $x$ ollessa moduloitu sinisignaali $Q[x]$ on hyvä approksimaatio $x$:n neliöllisestä verhokäyrästä. Toinen muuttuja on verhokäyrän ja sen derivaatan energioiden suhde eli

$$\alpha_1 = \frac{E\{\dot{V}^2\}}{E\{V^2\}}. \tag{37}$$

V on määritelty kaavassa (34) ja $\dot{V}$ on diskreetisti laskettu derivaatta. Kolmatta piirrettä varten määritetään muuttuja $D$, joka on peräkkäisten näytteiden erotus eli $D[x] = x_n - x_{n-1}$. Muuttuja $\alpha_2$ voidaan näin kirjoittaa

$$\alpha_2 = \frac{E\{(Dq[x])^2\}}{E\{q^2[x]\}} = \frac{E\{(Dq[x])^2\}}{E\{Q[x]\}}, \tag{38}$$

missä $q[x] = \sqrt{Q[x]}$. $Q[x]$ määriteltiin kaavassa (36). Neljäs parametri $\alpha_3$ on muotoa

$$\alpha_3 = \frac{E\{(DQ[x])^2\}}{(E\{Q[x]\})^2}. \tag{39}$$

### 9.1.2 Hetkittäisten ominaisuuksien vaihteluun perustuvat piirteet

Azzouz ja Nandi ovat esittäneet useita piirteitä analogisten ja digitalisten modulaatiolajien luokitteluun viitteessä [19]. Piirteet laskettiin signaalispektristä sekä hetkellisestä amplitudista, taajuudesta ja vaiheesta. Piirteitä käytetään analogisten AM-, FM-, DSB-, USB- ja LSB-signaalien sekä digitaalisten 2ASK-, 4ASK-, 2PSK-, 4PSK-, 2FSK-, ja 4FSK-signaalien luokitteluun.

Piirremuuttuja $\gamma_{max}$ on määritelty yhtälöllä

$$\gamma_{\max} = \max\left|DFT(A_{cn}(i))\right|^2 / N, \tag{40}$$

missä $A_{cn}(i)$ on normalisoidun keskitetyn hetkellisen amplitudin arvo ajan hetkellä $t = i/f_s$, ($i = 1, 2, \ldots, N$) . $A_{cn}(i)$ on määritelty

$$A_{cn}(i) = \frac{A(i)}{m_a} - 1, \tag{41}$$

missä $m_a$ on signaalijakson hetkellisen amplitudin keskiarvo eli

$$m_a = \frac{1}{N}\sum_{i=1}^{N} A(i). \tag{42}$$

Signaalin normalisointi on tarpeellinen, koska se kompensoi kanavan vahvistuksen. Muuttuja $\gamma_{max}$ esittää siis signaalin spektrin tehotiheyden maksimiarvon hetkellisen amplitudin suhteen laskettuna.

Toinen piirreparametri $\sigma_{ap}$ on hetkittäisen vaiheen epälineaarisen komponentin absoluuttisten arvojen keskihajonta, jossa otetaan huomioon vain signaalin voimakkaat jaksot. Voimakkaat jaksot on määritelty normalisoidun amplitudin kynnysarvon $t_a$ avulla eli vain ne näytteet otetaan huomioon, jolloin $A_n(i) > t_a$. Parametri $\sigma_{ap}$ määritellään seuraavasti:

$$\sigma_{ap} = \left[ \frac{1}{C} \left( \sum_{A_n(i)>t_a} \phi_{NL}^2(i) \right) - \left( \frac{1}{C} \sum_{A_n(i)>t_a} \left| \phi_{NL}(i) \right| \right)^2 \right]^{1/2}, \tag{43}$$

missä $\phi_{NL}(i)$ on hetkittäisen vaiheen epälineaarinen komponentti ajan hetkellä $t = i/f_s$. ja $C$ on niiden näytteiden $\phi_{NL}(i)$ määrä, jolloin $A_n(i) > t_a$.

Piirremuuttuja $\sigma_{dp}$ on muuten sama kuin edellä esitetty, mutta nyt hetkittäisen vaiheen epälineaarinen komponentti ei ole itseisarvo, vaan se lasketaan suoraan summatermiin eli

$$\sigma_{dp} = \left[ \frac{1}{C} \left( \sum_{A_n(i)>t_a} \phi_{NL}^2(i) \right) - \left( \frac{1}{C} \sum_{A_n(i)>t_a} \phi_{NL}(i) \right)^2 \right]^{1/2} \tag{44}$$

Piirremuuttuja $P$ mittaa spektrin symmetrisyyttä kantoaallon ympärillä ja se perustuu sivukaistojen spektrien tehoihin. $P$ saadaan kaavalla

$$P = \frac{P_L - P_U}{P_L + P_U}, \tag{45}$$

missä

$$P_L = \sum_{i=1}^{f_{cn}} \left| X_c(i) \right|^2, \tag{46}$$

$$P_U = \sum_{i=1}^{f_{cn}} \left| X_c(i + f_{cn} + 1) \right|^2, \tag{47}$$

ja $f_{cn}$ on kantoaaltotaajuutta vastaavan näytteen indeksi. $X_c(i)$ on signaalin $x(n)$ DFT:

$$X_c(i) = \sum_{n=0}^{N-1} x(n) e^{-jkn2\pi/N} \text{ ja } f_{cn} = \frac{f_c N}{f_s} - 1, \tag{48}$$

missä $N$ on näytteiden $x(n)$ määrä.

Piirremuuttuja $\sigma_a$ on normalisoidun hetkellisen amplitudin keskihajonta laskettuna ei-heikoista jaksoista. Piirre $\sigma_a$ on määritelty kaavalla

$$\sigma_a = \left[ \frac{1}{C} \left( \sum_{A_n(i) > t_a} A_{cn}^2(i) \right) - \left( \frac{1}{C} \sum_{A_n(i) > t_a} A_{cn}(i) \right)^2 \right]^{1/2}. \tag{49}$$

AM- ja MASK-signaalit voidaan erottaa toisistaan normalisoitujen hetkittäisten amplitudien kurtosiksella $\mu^a_{42}$, joka määritellään seuraavasti:

$$\mu^a_{42} = \frac{E\{A_{cn}^4(t)\}}{[E\{A_{cn}^2(t)\}]^2}. \tag{50}$$

## 9.2 Aallokepakettihajotelmaan perustuva piirreirrotus

Fourier- ja aallokemuunnos [20] ovat vanhoja käsitteitä, mutta signaalinkäsittelyn sovellusten myötä menetelmät ovat tulleet tärkeiksi ja ajankohtaisiksi. Fourier-muunnoksessa signaalit määritellään trigonometristen funktioiden lineaarikombinaatioina, kun taas aallokemuunnoksessa signaalin muodostavat osafunktiot ovat lokaaleja.

Aallokemuunnos saadaan suodattamalla muunnettava signaali analyysifunktion $\psi$ variaatioilla, jotka on skaalattu ja siirretty aika-avaruudessa. Analyysifunktio on muotoa

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left( \frac{t-b}{a} \right), \tag{51}$$

jossa $a$ on positiivinen reaaliluku. Muuttuja $a$ skaalaa aallokefunktion ja muuttuja $b$ siirtää sitä aika-avaruudessa. $1/\sqrt{a}$ on normalisointikerroin. Analyysifunktioiden joukkoa $\psi_{a,b}(t)$ kutsutaan kannaksi. Funktio $\psi_{a,b}(t)$ on ajallisesti lyhyt eli korkeataajuinen, kun $a < 1$ ja vastaavasti matalataajuuksinen, kun $a > 1$. Koska korkeataajuiset signaalit ovat

ajallisesti lyhyitä, aallokemuunnos analysoi signaalin pienet yksityiskohdat tarkemmalla resoluutiolla kuin suuret yksityiskohdat.

Kannan tulee olla täydellinen eli lineaarikombinaatioiden pitää pystyä kuvaamaan kaikki aikatason signaalit. Jos kanta on lisäksi ortogonaalinen, yhtään kantafunktiota ei voida muodostaa muiden funktioiden lineaarikombinaationa, ja tällöin aikatason signaalia vastaa yksikäsitteinen muunnettu signaali. Jatkuva aallokemuunnos (continuous wavelet transform, CWT) määritellään kaavalla

$$CWT_f(a,b) = \langle \psi_{a,b}(t), f(t) \rangle, \tag{52}$$

jossa $f(t)$ on muunnettava signaali, $\psi_{a,b}(t)$ analysoiva funktiojoukko ja sisätulo $\langle g,h \rangle$ määritellään seuraavasti:

$$\langle g,h \rangle = \int_{-\infty}^{\infty} g^*(t)h(t)dt \tag{53}$$

jossa $g^*(t)$ on funktion $g(t)$ kompleksikonjugaatti. Diskreetti aallokemuunnos tehdään yleensä suodattamalla ja skaalaamalla signaalia rekursiivisesti. Suodattavana funktiona käytetään edellä esitettyä analyysifunktiota. Esimerkiksi diskreetti Haarin aallokemuunnos vektorille $x$ määritellään siten, että muodostetaan suodattamalla kaksi vektoria $h$ ja $l$, joiden pituus on puolet signaalin $x$ pituudesta:

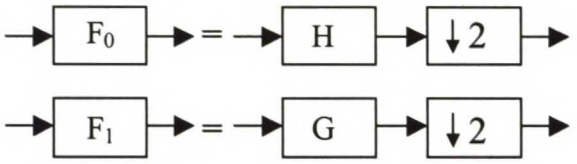
$$\begin{aligned} l[k] &= \frac{1}{\sqrt{2}}(x[2k] + x[2k+1]) \\ h[k] &= \frac{1}{\sqrt{2}}(x[2k] - x[2k+1]) \end{aligned} \tag{54}$$

Vektori $l$ on vektorin $x$ muotoinen signaali puolet pienemmällä resoluutiolla esitettynä eli signaalin $x$ matalataajuinen osa. Vastaavasti vektori $h$ on signaalin korkeataajuinen komponentti. Jakamalla vektorit $l$ ja $h$ edelleen rekursiivisesti matala- ja korkeataajuisen osaan päädytään lopulta signaalin $x$ moniresoluutioesitykseen eli aallokemuunnokseen.

Aallokepakettihajotelman (wavelet packet decomposition, WPD) voidaan ajatella olevan aallokemuunnoksen luonnollinen laajennus. WPD tuottaa signaalin kerroksittaisen hajotelman aika-alueesta taajuusalueeseen. Ylimmällä tasolla on signaalin aikaesitys. Laskettaessa kutakin tasoa ylhäältä alaspäin hajotelman hetkittäinen resoluutio pienenee

samanaikaisesti kuin taajuusresoluutio kasvaa. Hajotelmapuun alimmalla tasolla on signaalin taajuusesitys.

Seuraavassa esitetään Wickerhauserin [21] kehittämä menetelmä, jolla sovelletaan aallokemuunnosta WPD:aan. Merkitään $h(n)$:llä ja $g(n)$:llä äärellisen impulssivasteen omaavia alipäästö- ja ylipäästösuodattimia. Merkitään vektorilla $x(n)$ $N$ pituista alkuperäistä signaalia. Signaalin pituus $N$ on jonkun kokonaisluvun toinen potenssi. Lisäksi merkitään signaalin $x(n)$ ja suodattimien $h(n)$ ja $g(n)$ konvoluutiota ja desimointia kahdella symbolilla $F_0$ ja $F_1$. Konvoluutio ja desimaatio voidaan esittää WDP:ssa diskreettiaikaisena suodatuksena ja alinäytteistyksenä (kuva 18).




**Kuva 18.** Konvoluutio ja desimaatio WDP:ssa.

Suodatukselle ja desimaatiolle voidaan kirjoittaa yhtälöt

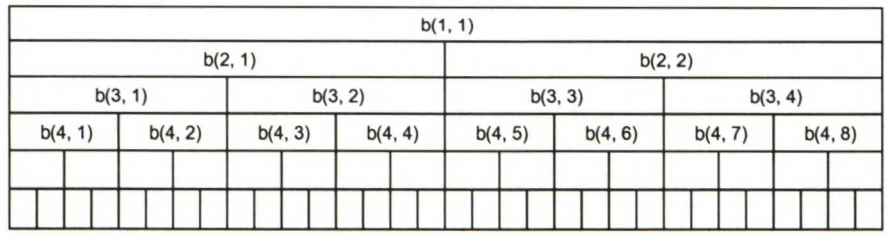
$$x_s(n) = F_0\{x(k)\} = \sum_k x(k)h(2n-k) \tag{55}$$

$$x_d(n) = F_1\{x(k)\} = \sum_k x(k)g(2n-k)\,. \tag{56}$$

Desimaatiosta johtuen $x_s(n)$ ja $x_d(n)$ sisältävät kukin puolet signaalin $x(n)$ näytemäärästä.

WPD lasketaan käyttäen suodatus-desimaatio –operaatiota rekursiivisesti. Jokainen rekursio luo uuden hajotelmatason. Täydellinen WPD voidaan esittää puumuodossa, jossa jokaisen haaran päässä on diskreetti sekvenssi. Haaran päässä olevaa sekvenssiä kutsutaan lokerovektoriksi (bin vector). Kunkin tason lokerovektorit lasketaan edellisen tason vektoreista käyttämällä funktioita $F_0$ ja $F_1$. Rekursion päätyttyä hajotelman alimmalla tasolla on enää yksi elementti jokaisessa lokerovektorissa.

Lokerovektorin sijainti puussa määritetään muodossa $b(l,c)$, jossa kukin lokero on indeksoitu kahdella parametrilla: taso $l$ ja sarake $c$. Kuvassa 19 on esitetty WPD-puun jokainen lokero edellä esitetyllä tavalla indeksoituna. Esimerkiksi $b(1,1)$ esittää ylintä tasoa eli alkuperäistä signaalia aikatasossa.




**Kuva 19.** WPD-puu.

Allokepakettihajotelmaa hyödyntävä piirrevektori voidaan muodostaa WPD-puun energiajakaumasta. Jokaisesta lokerovektorista lasketaan keskimääräinen energia $e_y$ kaavalla

$$e_y = \frac{1}{N}\mathbf{y}^T\mathbf{y}, \tag{57}$$

missä **y** on $N$-elementtinen vektori. Esimerkiksi viisitasoisesta hajotelmasta saadaan 63 energia-arvoa.

## 9.3 Perusteita lähetelajin ja sisällön luokittelevien piirteiden muodostamiseen

Sisällön mukaan signaaleita on luokiteltu ja tunnistettu viitteissä [22], [23] ja [24]. Seuraavassa esitetyt piirteet on laskettu audiosignaaleista. Piirteen laskentaa varten signaalit on pilkottu useaan samanmittaiseen osaan eli kehykseen. Kehykset voivat olla myös limittäisiä.

Viitteessä [24] on muodostettu viisi piirrettä musiikin ja puheen erottamiseen. Ensimmäinen piirre on matalanenergisten kehysten (low energy frame, LEF) osuus, joka on niiden kehysten määrä, joilla RMS-teho on alle puolet keskimääräisestä RMS-tehosta, suhteessa kaikkiin kehyksiin. Toinen piirre on spektrin putoamispiste (spectral roll-off point, SR). SR-piste on se taajuus, jonka alapuolella on 95% spektrin tehosta ja se lasketaan kaavalla

$$\sum_{f<SR} X[f] = 0.95 \sum_{f} X[f], \tag{58}$$

missä $X[f]$ on signaalin teho taajuudella $f$. Kolmas piirre on spektrivuo (spectral flux, SF), joka saadaan laskemalla kehyksen signaalispektrin vierekkäisten näytteiden erotusten summa eli

$$SF = \sum_k \left\| X[k] - |X[k+1]| \right\|. \tag{59}$$

Neljäntenä piirteenä laskettiin nolla-ylitys –nopeus (zero-crossing rate, ZC). Aikatasossa ZC-suhde on nollatason ylitysten määrä aikayksikössä. Viides piirre on spektrin keskipiste (spectral centroid, SC)

$$SC = \frac{\sum_k kX[f]}{\sum_k X[k]}, \tag{60}$$

missä $k$ on taajuutta $f$ vastaava indeksi.

Viitteessä [23] on muodostettu yhteensä 14 eri piirrettä siten, että kehys on jaettu alikehyksiin, joille on laskettu kahdeksan parametria: tehollisarvovolyymi (RMS volume), nolla-ylitys -nopeus, merkkitiheys, taajuuskeskipiste, kaistanleveys ja energiasuhde kolmelta alikaistalta. Piirteet on muodostettu näistä edellä mainituista parametreista tilastollisia suureita käyttäen.

Piirteiden muodostaminen signaalin sisällön analysoimiseksi on yllä mainituissa viitteissä tehty suoraan audiosignaaleista. Tässä työssä tarkasteltavana oleva mittausaineisto on kuitenkin vastaanottimen välitaajuudelta tallennettua signaalia. Edellä esitettyjen sisällön luokittelussa käytettyjen piirteiden soveltaminen mittausaineistoon ei ole relevanttia. Piirteiden soveltamiseen tarvitaan audiosignaaleja sisältävä havaintoaineisto.

# 10 Mittausaineiston analyysin tuloksia

Taulukossa 2 on esitetty analysoitavat signaalinäytteet. Signaalinäytteet on poimittu mittausaineistosta siten, että signaalityypit on mahdollisimman laajasti edustettuna. Analysoitavien signaalien taajuudet vaihtelevat noin 3,6 MHz:stä 1,7 GHz:iin. Näytteet

sisältävät seuraavia modulaatiolajeja: FM, AM, USB, LSB ja A1. Signaalien sisältö on pääosin puhetta, musiikkia tai dataa.

**Taulukko 2.** Analysoidut signaalinäytteet.

| Numero | Taajuus [MHz] | Kanava/lähetelaji | Modulaatio | Kaistanleveys [kHz] |
|---|---|---|---|---|
| 1 | 95,5000 | Radiomafia musiikki | FM | 30 |
| 2 | 95,5000 | Radiomafia puhe | FM | 30 |
| 3 | 103,3000 | Kiss FM musiikki | FM | 30 |
| 4 | 103,3000 | Kiss FM puhe | FM | 30 |
| 5 | 196,2550 | TV kanava 6 data | FM | 4000 |
| 6 | 196,2550 | TV kanava 6 data | FM | 4000 |
| 7 | 81,0000 | Taksiradio data | FM | 30 |
| 8 | 81,0000 | Taksiradio data | FM | 30 |
| 9 | 145,1750 | ra 2m kutsu | FM | 30 |
| 10 | 145,1750 | ra 2m puhe | FM | 30 |
| 11 | 164,5500 | vir puhe | FM | 30 |
| 12 | 164,5500 | vir puhe | FM | 30 |
| 13 | 113,7000 | majakka cw | USB | 2,5 |
| 14 | 113,7000 | majakka cw | USB | 2,5 |
| 15 | 131,9750 | siviili-ilm. puhe | AM | 8 |
| 16 | 131,9750 | siviili-ilm. puhe | AM | 8 |
| 17 | 318,0000 | ULA-linkki musiikki | FM | 30 |
| 18 | 318,0000 | ULA-linkki musiikki | FM | 30 |
| 19 | 391,3375 | Tetra-pääte data | FM | 30 |
| 20 | 391,3375 | Tetra-pääte data | FM | 30 |
| 21 | 463,1500 | NMT450 data | FM | 8 |
| 22 | 463,2500 | NMT450 data | FM | 8 |
| 23 | 1575,4200 | GPS-l1 data | FM | 8000 |
| 24 | 1575,4200 | GPS-l1 data | FM | 8000 |
| 25 | 1711,5000 | GSM1800 data | FM | 4000 |
| 26 | 1711,5000 | GSM1800 data | FM | 4000 |
| 27 | 9,9950 | aikamerkki cw | USB | 2,5 |
| 28 | 9,9950 | aikamerkki cw | USB | 2,5 |
| 29 | 21,5640 | Yleisradio puhe | AM | 8 |
| 30 | 21,5640 | Yleisradio puhe | AM | 8 |
| 31 | 3,6064 | ra puhe | LSB | 2,5 |
| 32 | 7,0437 | ra puhe | LSB | 2,5 |
| 33 | 21,1490 | ra-majakka cw | A1 | 2,5 |
| 34 | 21,1490 | ra-majakka cw | A1 | 2,5 |
| 35 | 4,2700 | pro numberdata | USB | 2,5 |
| 36 | 4,2700 | pro numberdata | USB | 2,5 |
| 37 | 14,0700 | ra data | USB | 2,5 |
| 38 | 14,0700 | ra data | USB | 2,5 |
| 39 | 3,5814 | ra data | USB | 2,5 |
| 40 | 3,5814 | ra data | USB | 2,5 |

## 10.1 Piirreirrotus ja datan esikäsittely

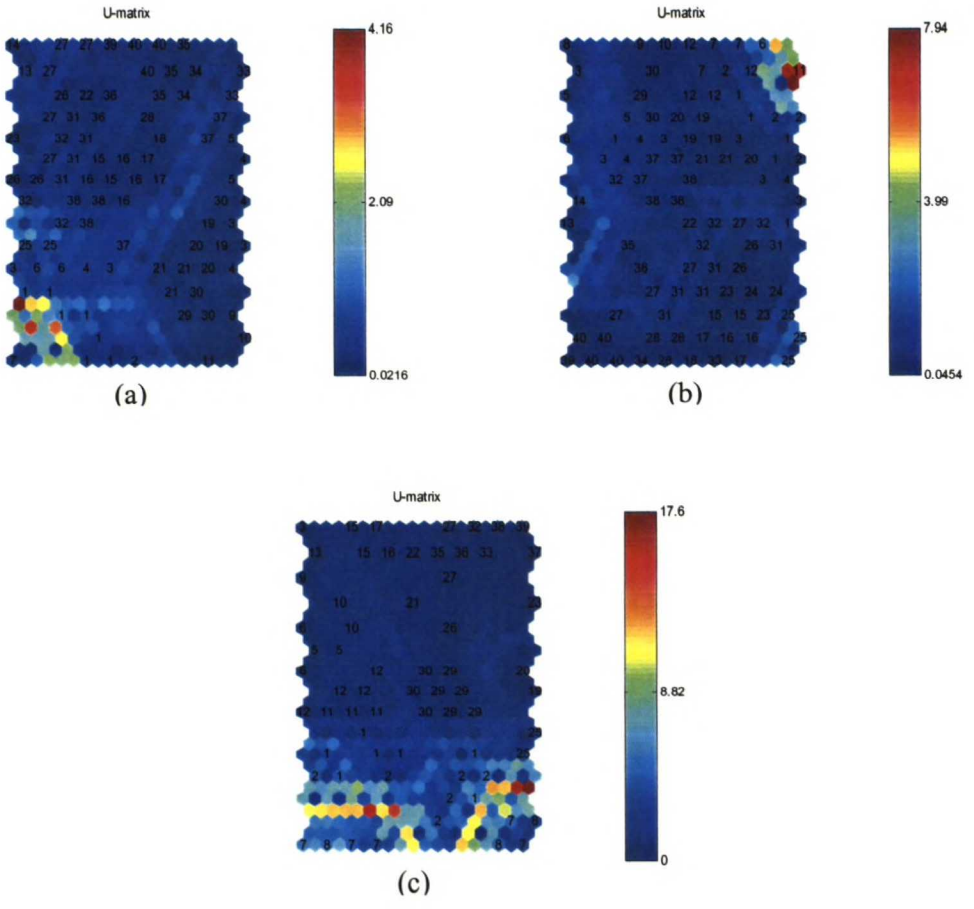
Edellä olevassa taulukossa olevista mittausnäytteistä laskettiin kolme eri datajoukkoa, joita käytettiin edellä kuvattujen analyysimenetelmien syötejoukkona. Jokainen näyte jaettiin kymmeneen osaan, joista jokaisesta laskettiin piirremuuttujat. Ensimmäinen syötematriisi **A** sisältää neljä piirrettä: $R_2$, $\alpha_1$, $\alpha_2$ ja $\alpha_3$. Matriisin kooksi saatiin näin 400 x 4 elementtiä.

Toinen syötematriisi **B** muodostettiin samalla tavalla, mutta nyt jokaisesta näytteen osasta laskettiin piirteet $\gamma_{max}$, $\sigma_{ap}$, $\sigma_{dp}$, $P$, $\sigma_a$ ja $\mu^a_{42}$. Piirreyhtälöissä esiintyvän normalisoidun amplitudin kynnysarvon $t_a$ optimaalinen arvo on välillä 0,99...1,05 [19]. Piirteiden arvot laskettiin käyttämällä arvoa $t_a$ = 1. Matriisin kooksi muodostui 400 x 6 elementtiä. Kolmas piirrematriisi **C** saatiin laskemalla jokaisesta näytteen osasta 5-tasoinen aallokepakettihajotelma, joka sisältää 63 lokerovektoria. Jokaisesta lokerovektorista laskettiin energia kaavalla (53), jolloin matriisin kooksi saatiin 400 x 63 elementtiä.

Edellä mainitut syötematriisit normalisoitiin siten, että jokaisen muuttujan (sarakkeen) keskiarvo on nolla ja varianssi yksi. Normalisoinnilla [2,25] varmistetaan se, että kaikilla komponenteilla on samansuuruinen vaikutus opetustulokseen. Skaalaus on tärkeää myös laskennallisessa merkityksessä. Jos arvot eroavat suuruudeltaan paljon, niiden aritmeettinen laskenta sisältää epätarkkuusriskin.

## 10.2 Analyysit itseorganisoituvaa karttaa soveltaen

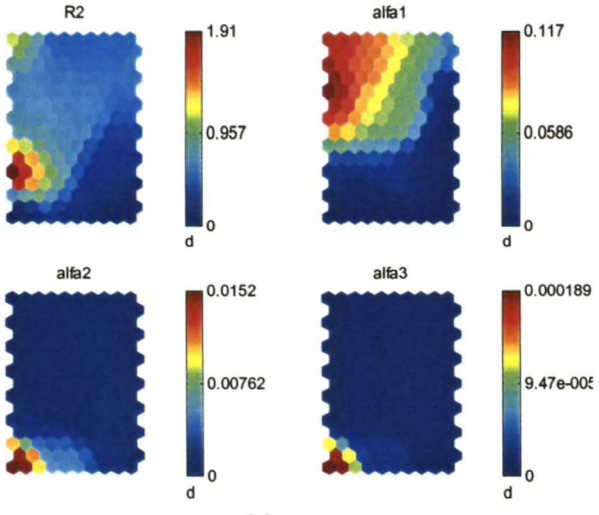
Itseorganisoituvan kartan avulla suoritetut analyysit ja visualisoinnit tehtiin Matlab-ohjelmistoon tehdyllä SOM Toolbox-sovelluksella [26]. Kuvassa 20 on esitetty kolme itseorganisoituvan kartan U-matriisia, joiden syötteenä on käytetty edellä mainittuja piirrematriiseja. Karttayksiköihin on lisäksi merkitty sen signaalinäytteen numero, jolle yksikkö on useimmin BMU.



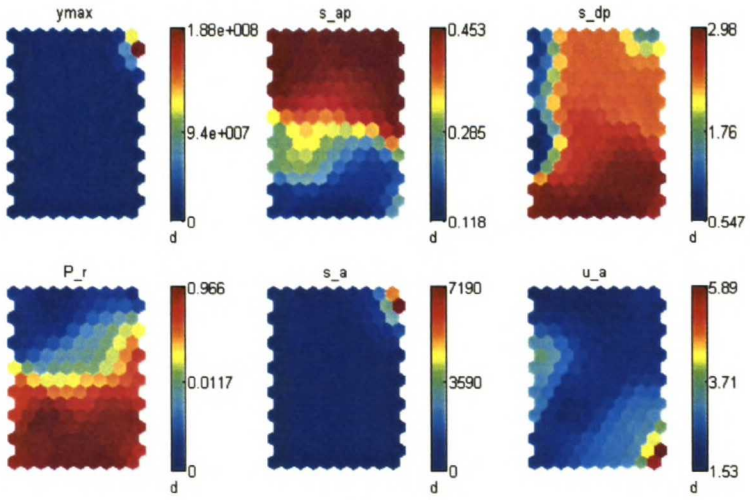

**Kuva 20.** Piirrematriiseista muodostetut U-matriisit: (a)-kohdassa on syötteenä matriisi **A**, (b)-kohdassa matriisi **B** ja (c)-kohdassa matriisi **C**. Numerot kertovat karttayksikön eniten saamien osumien luokan.

U-matriiseista nähdään, että signaalinäytteiden piirrevektorit sijaitsevat kartalla pääosin samalla alueella. Kartoista on vaikea erottaa selviä klustereita, mutta esimerkiksi kuvasta 20 (a) voidaan erottaa kolme tai neljä heikompaa klusteria. Kuvan 20 (b) kartassa näytteet 1, 2, 3 ja 4 ovat hajallaan kartalla. Kuvassa 20 (c) näytteet ovat edelleen hyvin hajallaan, mutta näytteet 29 ja 30 eli Yleisradion AM-lähetteet ovat klusteroituneet kartan keskelle.

Kuvassa 21 on U-matriisien komponenttitasot, joista nähdään kunkin piirremuuttujan vaikutus itseorganisoituvalla kartalla. Piirrematriisin **B** komponenttitasoja ei esitetä, koska 63 komponentin esittäminen samassa kuvassa ei ole havainnollista. Samalla alueella olevien komponenttien suuret arvot osoittavat muuttujien välistä korrelaatiota. Piirrematriisin **A** muuttujista $\alpha_2$ ja $\alpha_3$ näyttävät korreloivan voimakkaasti. Piirrematriisin **B** muuttujista voimakasta korrelaatiota osoittavat $\gamma_{max}$ ja $\sigma_a$.



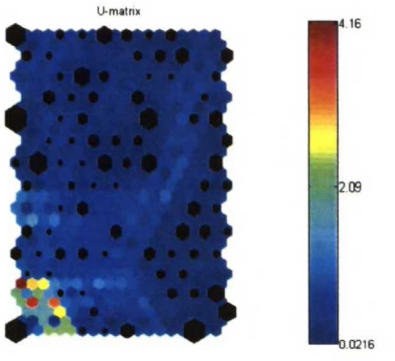

(a)




(b)

**Kuva 21.** Komponenttitasot kuvan 20 (a)- ja (b)-kohtien U-matriiseista.

Kuva 22 esittää U-matriisia, johon on merkitty mustilla monikulmioilla eniten osumia saaneet karttayksiköt, kun syötteenä on ollut piirrematriisi **A**. Mitä suurempi monikulmio on, sitä enemmän osumia kyseiselle karttayksikölle on tullut. Kuvasta voidaan erottaa viisi aluetta, jonne on tullut eniten osumia: molemmat alanurkat, keskellä, vasemmalla keskellä sekä ylhäällä vasemmalla

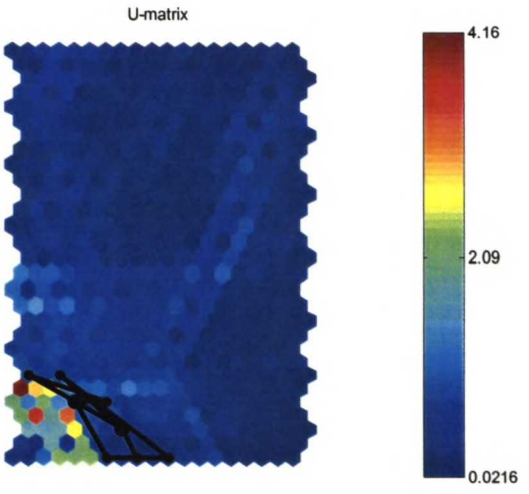



**Kuva 22.** SOM:n BMU:t eli eniten osumia keränneet karttayksiköt (matriisi **A**).

Syötedatan jakautumista kartalla voidaan tarkastella myös kuvan 23 tavoin. Kuvassa on esitetty komponenttien arvot jokaisessa karttayksikössä, kun syötteenä on piirrematriisi **A**. Kuvasta nähdään, että esimerkiksi vasemmassa ylänurkassa toisella komponentilla on suuri arvo. Toisin sanoen $\alpha_1$-parametrin suurilla arvoilla sijoitutaan kartan vasempaan yläkulmaan.

**Kuva 23.** Komponenttien arvot karttayksiköittäin (syötematriisi **A**).

Näytedatan piirrevektoreiden ajallista sijoittumista itseorganisoituvalle kartalle havainnollistetaan kuvan 24 esimerkillä. Kuvasta nähdään, että ensimmäisen näytesignaalin trajektori sijaitsee kartan vasemmassa alakulmassa eli samalla alueella kuin kuvan 20 kartalla olevat ensimmäisen näytesignaalin BMU:t.




**Kuva 24.** Piirrematriisin **B** ensimmäisen signaalinäytteen piirrevektoreiden trajektori.

Kaksiulotteinen itseorganisoituva kartta voidaan jakaa alueisiin erilaisilla menetelmillä [27]. Davies-Bouldin -indeksi perustuu klustereiden koon ja niiden välisten etäisyyksien laskentaan. Indeksin mukaan optimaalisin klusterimäärä saavutetaan silloin, kun indeksi saa pienimmän arvonsa. Indeksi $i_{DB}$ lasketaan kaavalla

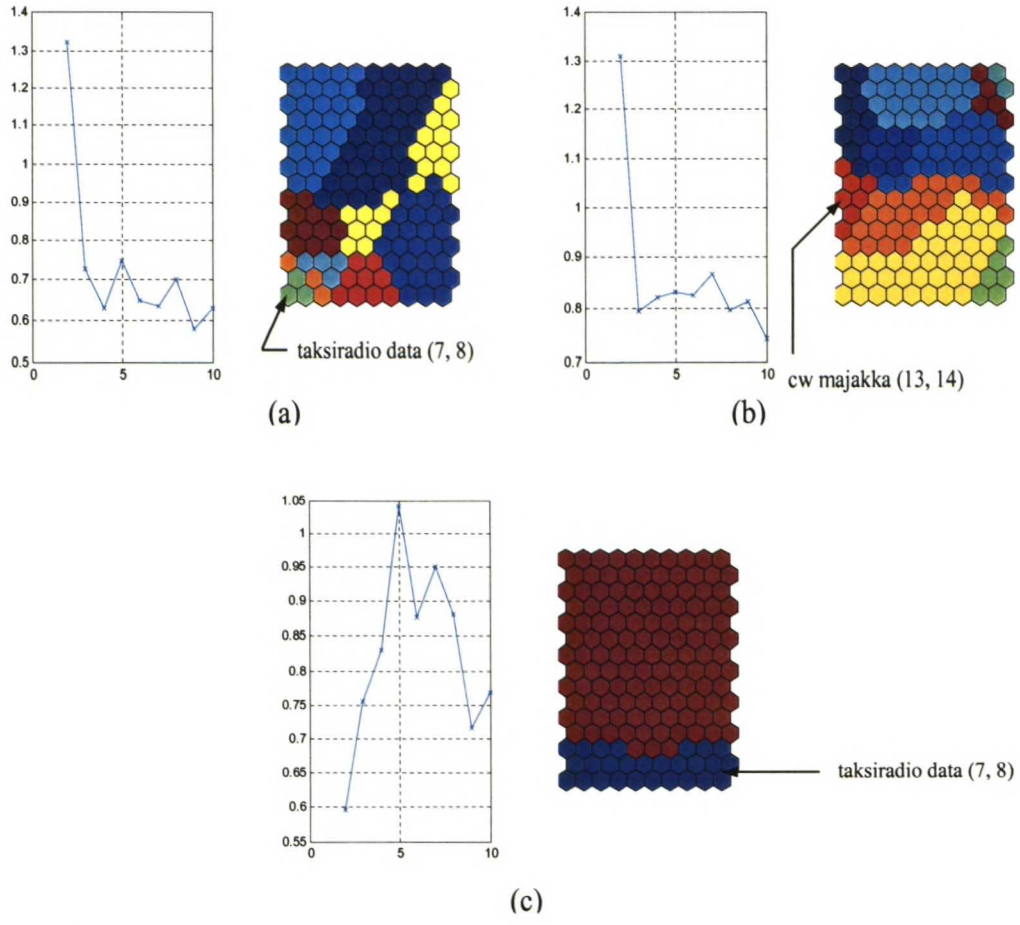
$$i_{DB} = \frac{1}{C}\sum_{k=1}^{C} \max_{l \neq k}\left\{\frac{S_c(Q_k) + S_c(Q_l)}{d_{ce}(Q_k, Q_l)}\right\}, \tag{61}$$

missä $Q$ on klusteri, $C$ on klustereiden määrä, $S_c$ on keskipiste-etäisyys ja $d_{ce}$ on klustereiden keskipisteiden välinen etäisyys. Keskipiste-etäisyys ja keskipisteiden välinen etäisyys lasketaan kaavoista

$$S_c = \frac{\sum_i \|x_i - c_k\|}{N_k} \tag{62}$$

$$d_{ce} = \|c_k - c_l\|. \tag{63}$$

Muuttujat $c_k$ ja $c_l$ ovat klustereiden $Q_k$ ja $Q_l$ keskipisteet, $x_i$ on näytevektori ja $N_k$ klusterissa olevien näytteiden määrä. Kuvaan 25 on laskettu itseorganisoituvan kartan Bouldin-Davies -indeksit, kun klustereita on yhdestä kymmeneen kappaletta. Kuvaan on lisäksi piirretty optimaalisesti klusteroidut kartat. Optimaaliset klusterimäärät ovat 9, 10 ja 2 klusteria. Piirrematriisin **A** kohdalla on huomattava, että alueet eivät ole yhtenäisiä ja piirrematriisin **C** kartta jakaantuu vain kahteen alueeseen.

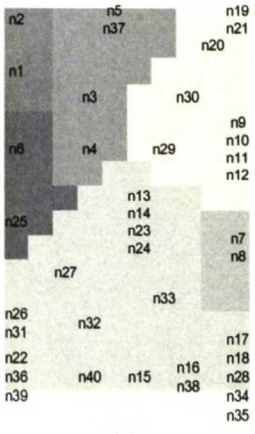



**Kuva 25.** Davies-Bouldin -indeksi ja klusteroitu SOM, syötteenä matriisit A (a), B (b) ja C (c).

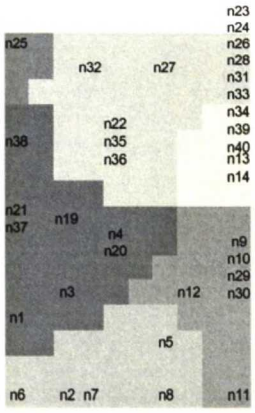
Verrattaessa kuvia 25 ja 20 on vaikea päätellä tarkasti, mitkä signaalinäytteet kuuluvat mihinkin klusteriin, koska U-matriisissa osa näytteistä on sijoittunut hajalleen kartalla. Kuitenkin osa klustereista voidaan identifioida U-matriisin perusteella. Kuvaan on manuaalisesti lisätty muutaman klusterin sisältö.

Kuvassa 26 on esitetty histogrammikartta (histogram-map) [28], johon on merkitty jokaisen näytesignaalin BMU. Histogrammikartta on laskettu käyttämällä syötteenä edellä esitettyjä klusteroituja karttoja. Histogrammikartan jokainen karttayksikkö kuvaa näytevektoreiden jakautumista klustereiden välillä piirreavaruudessa.

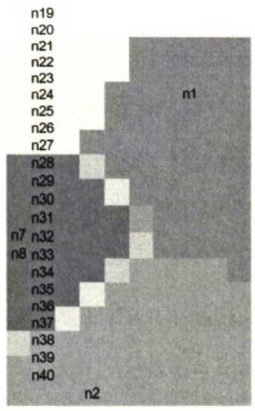
Histogrammikartan syötedata on muodostettu siten, että jokaisen klusterin osumien määrä muodostaa syötematriisin rivin. Näin opetusdatan kooksi muodostui 40 x 9, 40 x 10 ja 40 x 2 elementtiä. Histogrammikartat on klusteroitu edellä esitettyä Davies-Bouldin -indeksiä käyttäen.




(a)




(b)



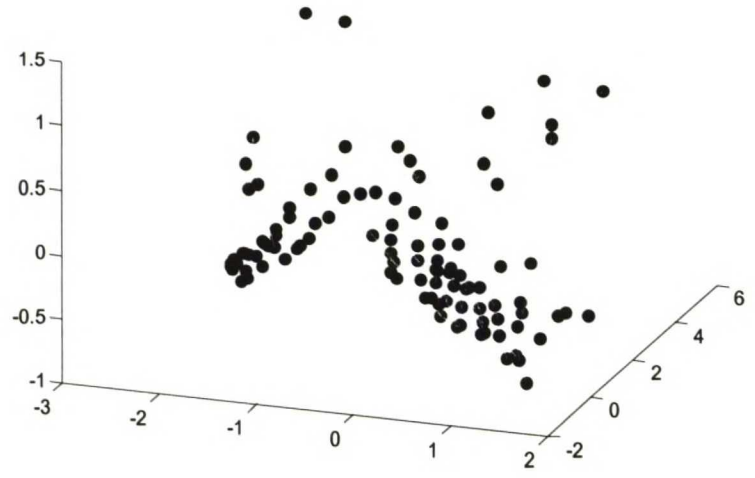

(c)

**Kuva 26.** Histogrammikartat eri piirrematriiseilla muodostettuna.

Kuvasta 26 (a) nähdään, että osa signaalityypeistä on sijoittunut omaan klusteriinsa ja osa näytteistä on jakautunut useaan klusteriin. Signaalinäytteet 1 ja 2 (Radiomafia FM) muodostavat oman alueensa ja näytteet 7 ja 8 (taksiradio data FM) samoin. Huomattava on myös, että suurin osa näytepareista (sama signaali) sijaitsee kartalla vierekkäin. Kuvassa 26 (b) suurin osa näytteistä sijoittuu oikealla keskellä olevaan klusteriin. Näytepa-

rien BMU:t eivät ole vierekkäin kuin osassa näytteitä. Kuvasta 26 (c) huomataan, että aallokepakettihajotelman avulla lasketut piirteet eivät erottele näytteitä järkevästi. Kuvan 26 mukaan selvää klusteroitumista modulaation, taajuuden tai lähetetyypin mukaan ei ilmene.

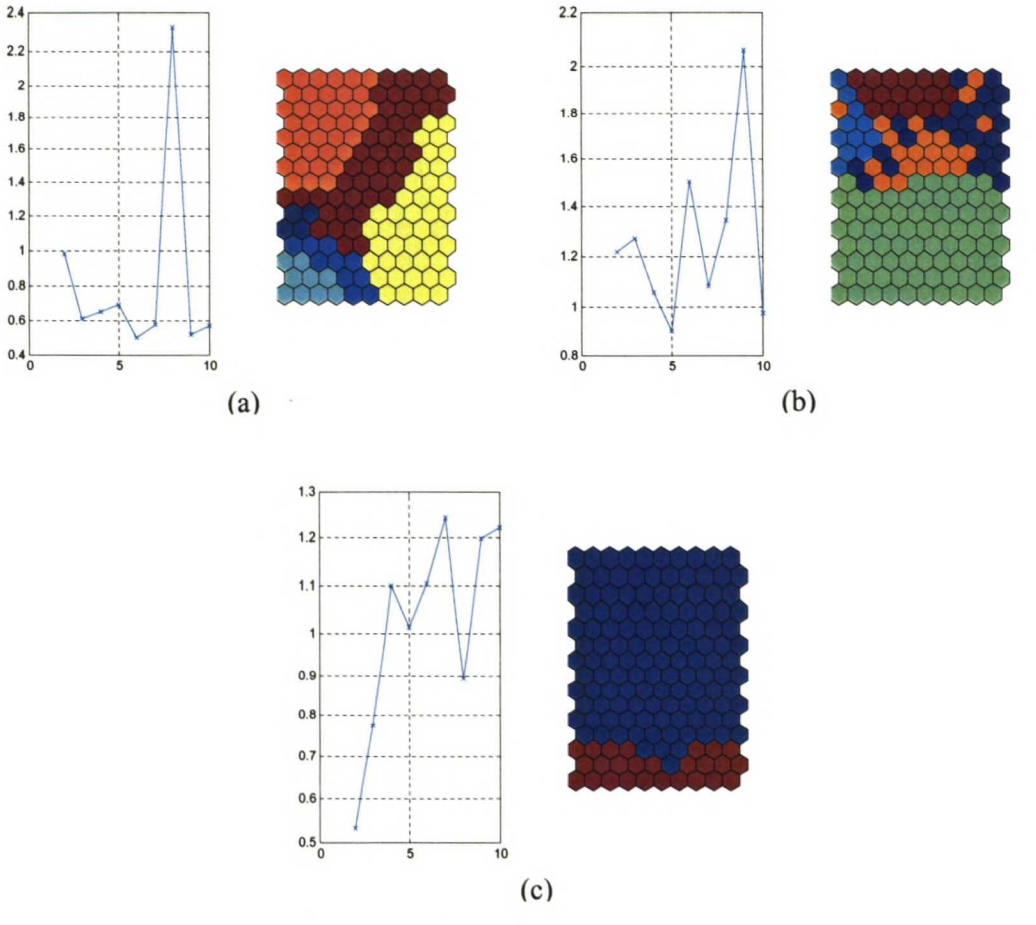
Kuvassa 27 on esimerkki Sammonin kuvaksen käytöstä. Kuvan 20 (a) itseorganisoituvan kartan mallivektorit on kuvattu kolmiulotteisesti. Kuvasta voidaan havainnoida syötedatan rakenne. Kuvasta voidaan havaita kaksi selvää tihentymää.




**Kuva 27.** Sammonin kuvaus matriisin **A** SOM:sta.

## 10.3 Oppiva vektorikvantisointi

Oppivaa vektorikvantisointia on tässä käytetty itseorganisoituvan kartan mallivektoreiden hienosäätöön. Algoritmin iteraatiokertojen määräksi asetettiin 400 ja opetuskertoimen arvoksi laitettiin 0.05. Hienosäädetyt kartat klusteroitiin käyttäen edellä esitettyä Davies-Bouldin -indeksiä. Käytettäessä syötejoukkona piirrematriisia **A** kartan optimaalinen klusteriluku on kuusi. Piirrematriisia **B** käytettäessä klustereiden optimimäärä on viisi, mutta kaikki klusterit eivät ole yhtenäisiä kuten kuvasta 28 nähdään. Syötteen ollessa piirrematriisi **C** tulos on saman kaltainen kuin ilman LVQ-algoritmia.




**Kuva 28.** LVQ:lla muokatut SOM:t.

## 10.4 kNN-luokitin

Lähimmän naapurin luokittelua varten piirrevektorit jaettiin opetusjoukkoon ja testinäytteisiin. Opetusjoukko jaettiin 14 eri luokkaan taajuusalueen, modulaation ja lähetelajin mukaan. Taulukossa 3 on esitetty opetusjoukon näytteiden luokat. Jokaisessa luokassa on opetusjoukon vektoreita 10 –30 kappaletta.

**Taulukko 3.** Opetusjoukon luokat.

| Luokka | Näytteet | Luokka | Näytteet |
|---|---|---|---|
| 1 | 2, 4 | 8 | 20, 22 |
| 2 | 6 | 9 | 24, 26 |
| 3 | 8 | 10 | 28, |
| 4 | 10, 12 | 11 | 30 |
| 5 | 14 | 12 | 32 |
| 6 | 16 | 13 | 34 |
| 7 | 18 | 14 | 36, 38, 40 |

Piirrevektorit luokiteltiin lähimmän naapurin menetelmällä valitsemalla k:n arvoksi 1, 3, 5 ja 7. Taulukossa 4 on esitetty luokittelun tulokset. Paras tulos saavutettiin piirrematriisin C luokittelussa, jolloin oikein luokiteltiin 81,5 % näytteistä. Tulos saavutettiin *k*:n arvoilla 1 ja 5 kuten taulukosta nähdään. Luokkakohtaiset tulokset ovat liitteessä 1.

**Taulukko 4.** kNN-luokittelun tulokset.

| Piirrematriisi | Luokittelutulos [%] *k*:n arvo | | | |
|---|---|---|---|---|
| | 1 | 3 | 5 | 7 |
| A | 77,0 | 77,0 | 76,5 | 75,5 |
| B | 74,0 | 76,5 | 73,5 | 72,0 |
| C | 81,5 | 82,0 | 81,5 | 80,5 |

## 10.5 Monikerrosperseptroni

Monikerrosperseptroni-neuroverkkoa käytettiin signaalien luokitteluun syötejoukon ollessa edellä esitetyt kolme piirrematriisia. Neuroverkko koostui kahdesta piilokerroksesta ja ulostulokerroksesta. Kaikissa kerroksissa aktivaatiofunktiona käytettiin sigmoid-funktiota. Neuroverkon kytkentöjen painot alustettiin satunnaisesti. Puolet piirrevektoreista käytettiin verkon opettamiseen ja puolet luokittelun testaamiseen. Opetusvaiheessa verkko opetettiin batch-algoritmilla eli opetusjoukko syötettiin neuroverkolle kokonaisuudessaan, ennen kuin painoja päivitettiin. Päivitykseen käytettiin takaisinvirtausalgoritmia. Iteraatiokierroksia oli 1000 ja opetuskertoimen arvo oli 0,05.

Neuroverkon toimintaa simuloitiin Matlab-sovelluksen Neural Network Toolbox -laajennuksella [29]. Luokittelutulokset on esitetty taulukossa 5. Ulostulokerroksessa oli 14 neuronia eli saman verran kuin on luokkia. Piilokerroksien neuronien määrä selviää taulukosta. Yksittäisten luokkien luokittelutulokset ovat liitteessä 2.

**Taulukko 5.** MLP-luokittelun tulokset.

| Piirrematriisi | Neuronien lkm piilokerroksissa | Tulos [%] |
|---|---|---|
| A | 20 ja 30 | 44,5 |
| A | 50 ja 80 | 55,0 |
| B | 20 ja 30 | 54,5 |
| B | 50 ja 80 | 62,0 |
| C | 20 ja 30 | 51,0 |
| C | 50 ja 80 | 51,5 |

Luokittelua pyrittiin parantamaan käyttämällä pääkomponenttianalyysia. Piirrematriisin **C** opetusjoukosta laskettiin kahdeksan pääkomponenttia, jotka muodostivat neuroverkon syötedatan. Verkon luokittelukyky testattiin käyttäen piirrematriisin testijoukkoa, josta laskettiin pääkomponentit opetusdatan muunnosmatriisilla. Taulukko 6 on luokittelun tulokset.

**Taulukko 6.** MLP-luokittimen tulokset: PCA.

| Piirrematriisi | Neuronien lkm piilokerroksissa | Tulos [%] |
|---|---|---|
| C | 20 ja 30 | 38,0 |
| C | 50 ja 80 | 44,5 |

# 11 Yhteenveto ja johtopäätökset

Tässä diplomityössä sovellettiin informaatiotekniikan menetelmiä radiotaajuisen mittausaineiston analyysiin ja visualisointiin. Tarkoituksena oli tutkia ja demonstroida eri menetelmiä hahmontunnistusjärjestelmän kehittämisen eri vaiheissa. Lisäksi tarkoituksena oli tuottaa Matlab-koodia, jota voidaan hyödyntää muille havaintoaineistoille.

Tutkittuja menetelmiä olivat itseorganisoituva kartta, pääkomponenttianalyysi, lähimmän naapurin menetelmä, monikerrosperseptroni-neuroverkko ja oppiva vektorikvantisaatio. Hahmontunnistusjärjestelmän suunnittelusta ja edellä mainituista menetelmistä esitettiin perusteet, jonka jälkeen tarkasteltiin piirteiden muodostamista mittausaineistosta. Piirteiden laskentaan käytettiin kolmea eri tapaa: verhokäyrän varianssin ja neliöllisen keskiarvon suhde, hetkittäisten ominaisuuksien vaihteluun perustuvat piirteet ja aallokepakettihajotelma. Lopuksi muodostettuja piirrematriiseja käytettiin syötteenä valituille menetelmille.

Hahmontunnistusjärjestelmän rakentaminen on monimutkainen prosessi. Havaintoaineiston prosessoituminen luokittelu- tai mallinnustulokseksi on monivaiheinen ketju. Tässä työssä tutkittiin, miten eri menetelmiä voidaan käyttää luokitinjärjestelmän suunnittelussa. Ominaispiirteensä aiheutti mittausaineisto, joka oli digitaalisesti välitaajuista RF-signaalia. Signaaleissa olevia häiriöitä ei tutkittu millään tavalla, mikä varmasti vai-

kutti lopputulokseen. Toisaalta tutkitut signaalit kattoivat vain osan sähkömagneettisen spektrin alueesta.

Itseorganisoituvaa karttaa voidaan käyttää moniulotteisen datan esittämiseen ja myös luokitteluun. Muodostetut U-matriisit osoittavat, että SOM:lla pystytään visualisoimaan signaalinäytteiden piirrevektoreita. Erityyppiset signaalit eivät kuitenkaan erotu U-matriisissa niin hyvin kuin olisi voinut odottaa. Yksi syy on käytetyt piirteiden laskentamenetelmät. Valitut menetelmät eivät tulosten perusteella sovellu riittävän hyvin tämän tyyppisen mittausaineiston käsittelyyn. Näin U-matriisia voidaankin käyttää hahmontunnistusjärjestelmää suunniteltaessa piirteiden testaamiseen ja esittämiseen. Toinen syy voi olla kartan alustaminen satunnaisesti.

SOM:n komponenttitasojen avulla voidaan tutkia piirteiden välisiä korrelaatioita. Samalla alueella kartalla esiintyvät komponentit osoittavat suurta korrelaatiota. BMU- ja trajektoriesityksillä voidaan edelleen tarkastella piirrevektoreiden jakaantumista ja kulkua piirreavaruudessa. Trajektori on käyttökelpoinen väline, kun halutaan selvittää piirrevektorin liikettä ajallisesti. Signaalin yksiselitteinen piirre ilmenee pienellä alueella liikkuvana trajektorina.

Itseorganisoituvaa karttaa voidaan käyttää myös luokittelun tekemiseen. Kartta jaetaan jollakin metodilla alueisiin eli klustereihin, jonka jälkeen näytevektori voidaan tuoda kartalle luokitusta varten. SOM:n käyttö luokitteluun edellyttää, että signaalinäytteistä lasketut piirteet ovat riittävän erottelevia. Histogrammikartan avulla voidaan tutkia piirrevektoreiden jakautumista eri klustereiden välillä. Oppiva vektorikvantisointi soveltuu esimerkiksi itseorganisoituvan kartan mallivektoreiden hienosäätöön. Vektorikvantisaatio vähentää klustereiden määrää SOM:ssa ainakin piirrematriisien **A** ja **B** kohdalla. Syötejoukon ollessa kolmas piirrematriisi havaittavia muutoksia ei ole.

Luokittelumenetelmiä, kNN ja MLP-verkko, voidaan käyttää mittausaineiston luokitteluun. Tuloksien perusteella kNN soveltuu paremmin tämän kaltaisten piirrevektoreiden luokitteluun. Luokittelutulos ei kuitenkaan ole täysin oikea, vaan osa tiettyjen signaalien piirrevektoreista luokitellaan väärin. kNN:ssä luokittelutuloksen ratkaisee opetus- ja testivektoreiden samankaltaisuus ja piirrevektoreiden separoituvuus toisistaan. Jos vektorit ovat lomittuneet runsaasti, lähimmän naapurin luokka voi vaihdella huomattavasti.

Myöskään MLP-verkko ei tuottanut virheetöntä luokittelua. MLP:ssä on useita parametreja, jotka voivat vaikuttaa lopputulokseen: aktivaatiofunktion valinta, painojen alustaminen, piilokerrosten määrä, neuroneiden määrä eri kerroksissa ja harhan (bias) suuruus. Parametreja säätämällä verkko kykenee kuitenkin mallintamaan monimutkaisiakin epälineaarisuuksia. Pääkomponenttianalyysilla voidaan pienentää syötedatan dimensiota. Tässä tapauksessa PCA:n käyttö ei parantanut luokittelun tulosta. Suurimpana syynä lienee piirteiden valinta radiotaajuisesta mittausaineistosta.

Edellä esitettyjä informaatiotekniikan menetelmiä voidaan käyttää, kun rakennetaan taajuushallinnan ja -monitoroinnin apuvälineitä. Erittäin huolellinen on oltava RF-signaalista laskettavien piirteiden valinnassa. Hyvin separoiva piirremuuttuja on perusedellytys kehitettäessä luokittelu- ja visualisointijärjestelmiä.

## 11.1 Jatkotutkimus

Tulevaisuudessa olisi hyödyllistä tutkia muiden piirteenlaskentamenetelmien soveltuvuutta radiotaajuisen mittausaineiston käsittelyyn. Lisäksi tässä työssä käytettyjä piirteitä voisi yrittää yhdistää tai muokata jollakin menetelmällä. Aallokepakettihajotelman lokerovektoreista kannattaisi laskea muitakin muuttujia kuin pelkkä energia.

Itseorganisoituvan kartan hienosäätö ja alustaminen kaipaavat lisätutkimista. Kartan klusteroinnissa voisi käyttää muitakin menetelmiä kuin Davies-Bouldin –indeksiä. Monikerrosperceptroni-verkon rakenteen ja aktivaatiofunktion muuttaminen sekä parametrien säätäminen parantaisi varmasti luokittelutuloksia.

Mielenkiintoista olisi tutkia kattavampaa mittausaineistoa, jossa esimerkiksi taajuusalue olisi laajempi. Välitaajuisen mittausaineiston tilalla voisi käyttää radiotaajuista signaalia tai sisältöä tutkittaessa mittausaineisto olisi demoduloitu eli se olisi audiosignaalia. Signaalin häiriöihin ei otettu kantaa tässä tutkimuksessa. Häiriöiden ja niiden suodattamisen vaikutusta piirrelaskentaan ja itse analyysin voisi tarkastella.

# Viitteet


[1] *Sotatekninen arvio ja ennuste*, STAE 2005, Sotatalousosasto, Pääesikunta, 2004.

[2] T. Kohonen: *The Self-Organizing Maps*. 3rd Edition, Springer-Verlag, Berlin, Heidelberg, 2001.

[3] D. Hand, H. Mannila ja P. Smyth: *Principles of Data Mining*. The MIT Press, 2001.

[4] P.-L. Forsström ja K. Vasko: *Neula heinäsuovassa? Tiedon louhintaa ja laskennallista älykkyyttä Suomessa.* Raportti, CSC - Tieteellinen laskenta Oy, 2001. http://www.csc.fi/selvitykset2001/datamining/

[5] S. Haykin: *Neural networks, a comprehensive foundation*. Prentice-Hall Inc, 1999.

[6] S. Theodoridis ja K. Koutroumbas: *Pattern Recognition*. Second Edition, Academic Press, 2003.

[7] V. A. Niskanen: *Sumea logiikka – kirkasta älyä ja mallinnusta*. WSOY, 2003.

[8] J. Vesanto: Data Mining Techniques Based on the Self-Organizing Map. Diplomityö, TKK, 1997.

[9] F. Mulier ja V. Cherkassky: *Learning rate schedules for self-organizing maps.* In Proc. of the 12 International Conference on Pattern Recognition, sivut 224 – 228, 1994.

[10] A. Ultsch ja H. Siemon: *Technical Report 329*. University of Dortmund, Dortmund, Germany 1989.

[11] M. Kasslin, J. Kangas ja O. Simula: *Process state monitoring using self-organizing map*. Artificial Neural Networks 2, volume II, sivut 1531-1534, Amsterdam, Netherlands, 1992.

[12] O. Simula, P. Vasara, J. Vesanto ja R.-R. Helminen: *The Self-Organizing Map in Industry Analysis*, Laboratory of Computer and Information Science, HUT, 1997.

[13] S. P. Luttrell: *Hierarchical self-organizing networks*, In Proceedings of the ICANN'89, 1989.

[14] E. Oja: *Subspace methods of pattern recognition.* Volume 6 of Pattern recognition and image processing series, John Wiley & Sons, 1983.

[15] A. Lehto ja A. Räisänen: *RF- ja Mikroaaltotekniikka*. Otatieto, 2001.

[16] A. Oppenheim ja R. Schafer: *Discrete Time Signal Processing*. Prentice-Hall Inc, New Jersey, 1989.

[17] Y.T. Chan ja L.G. Gadbois: *Identification of the modulation type of a signal*. Signal Processing, Vol. 16, no. 3, pp.149-154, Feb. 1989.

[18] I. Druckmann, E. I. Plotkin ja M. N. S. Swamy: *Automatic Modulation Type Recognition*. IEEE, 1998.

[19] E.E. Azzouz ja A. K. Nandi: *Procedure for automatic recognition of analogue and digital modulations*. IEE Proceedings – Communication, vol. 143, no. 5, Oct. 1996.

[20] M. Vettereli ja J. Kovacevic: *Wavelets and Subband Coding*. Prentice-Hall Inc., New Jersey, 1995.

[21] M. V. Wickerhauser: *Lectures on wavelet packet algorithms*. Technical report, Washingto University, Department of Mathematics, 1992.

[22] J. Sanders: *Real time discrimination of broadcast speech/music.* In proceeding oh the ICASSP 993-996, 1996.

[23] E.M. Saad, M.I. El-Adawy, Abu-El-Wafa ja A. A. Wahba: *A Multifeature Speech/Music Discrimination System*. 19th National Radio Science Conference, Alexandria, March, 19-21 2002.

[24] Z. Liu, J. Huang ja Y. Wang: *Classification of TV Programs Based on Audio Information Using Hidden Markov Model.* Multimedia Signal Processing, IEEE Second Workshop on 7-9 Dec. 1998. Sivut: 27 - 32.

[25] J. Himberg, J. Ahola, E. Alhoniemi, J. Vesanto ja O. Simula: *The Self-Organizing Map as a Tool in Knowledge Engineering*. HUT, 1999.

[26] J. Vesanto, J. Himberg, E. Alhoniemi, ja J. Parhankangas: *SOM Toolbox for Matlab 5. Versio 2.0.* Informaatiotekniikan laboratorio, TKK, 2000.

[27] J. Vesanto ja E. Alhoniemi: *Clustering of the self-organizing map.* IEEE Transactions on Neural Networks, 11(3): 586-600, May 2000.

[28] P. Lehtimäki, K. Raivio, ja O. Simula: *Mobile Radio Access Network Monitoring Using the Self-Organizing Map.* In Proceedings of the European Symposium on Artificial Neural Networks (ESANN), pages 231-236, Bruges, 2002.

[29] *Neural Network Toolbox 4.0.1 for Matlab R12.* The MathWorks, 2000.

# kNN-luokittelun tulokset ($k = 1$)

Piirrematriisi **A**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 15 | 1 | | | | | | 1 | | | | 3 | | |
| | 2 | 3 | 7 | | | | | | | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | | 1 | | 19 | | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 10 | | | | | | | | |
| | 7 | | | | | | | 9 | | | 1 | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | 1 | | | | | | | | 14 | | | 5 | | |
| | 10 | | | | | | | | | | | | 10 | | |
| | 11 | 1 | | | | | | | | | | 9 | | | |
| | 12 | | | | | | | | | 3 | 1 | | 6 | | |
| | 13 | | | | | | | 6 | | | 4 | | | | |
| | 14 | 5 | | | | | | | | | | | | | 25 |

Piirrematriisi **B**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 16 | | 1 | | | | | 1 | | | | 2 | | |
| | 2 | 2 | 8 | | | | | | | | | | | | |
| | 3 | | | 4 | 6 | | | | | | | | | | |
| | 4 | | 1 | | 18 | | | | | | | 1 | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 9 | | | 1 | | | | | |
| | 7 | | | | | | | 9 | | | 1 | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | 1 | | | | | | | | 16 | | | 3 | | |
| | 10 | | | | | | | | | | | | 10 | | |
| | 11 | 1 | | | | | | | | | | 9 | | | |
| | 12 | | | | | | | | | 4 | 1 | | 5 | | |
| | 13 | | | | | | | 10 | | | | | | | |
| | 14 | | | | | | | | | | | | 4 | | 26 |

Piirrematriisi **C**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 18 | | | 2 | | | | | | | | | | |
| | 2 | | 10 | | | | | | | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | 10 | | | 10 | | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 10 | | | | | | | | |
| | 7 | | | | | | | 10 | | | | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | | 2 | | | | | | | 15 | | | | | |
| | 10 | | | | | | | | | | 10 | | | | |
| | 11 | | | | | | | | | | | 10 | | | |
| | 12 | | | | | | | | | | | | | | 10 |
| | 13 | | | | | | | | | | | | | 10 | |
| | 14 | | | | | | | | | | 10 | | | | 20 |

# kNN-luokittelun tulokset ($k = 3$)

Piirrematriisi **A**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 16 | 3 | | | | | | 1 | | | | | | |
| | 2 | 3 | 7 | | | | | | | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | | | | 20 | | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 10 | | | | | | | | |
| | 7 | | | | | | | 9 | | | 1 | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | | | | | | | | | 14 | | | 6 | | |
| | 10 | | | | | | | | | | | | 10 | | |
| | 11 | | | | | | | | | | | 10 | | | |
| | 12 | | | | | | | | | 6 | 1 | | 3 | | |
| | 13 | | | | | | | 10 | | | | | | | |
| | 14 | 5 | | | | | | | | | | | | | 25 |

Piirrematriisi **B**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 18 | | 1 | | | | | 1 | | | | | | |
| | 2 | | 8 | | | | | | | | | 2 | | | |
| | 3 | 1 | | 4 | 5 | | | | | | | | | | |
| | 4 | | | | 19 | | | | | | | 1 | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 9 | | | 1 | | | | | |
| | 7 | | | | | | | 9 | | | 1 | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | | | | | | | | | 16 | | | 4 | | |
| | 10 | | | | | | | | | | | | 10 | | |
| | 11 | | | | | | | | | | | 10 | | | |
| | 12 | | | | | | | | | 5 | 2 | | 3 | | |
| | 13 | | | | | | | 10 | | | | | | | |
| | 14 | | | | | | | | 2 | | | | | | 28 |

Piirrematriisi **C**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 17 | | | 3 | | | | | | | | | | |
| | 2 | | 10 | | | | | | | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | 10 | | | 10 | | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 10 | | | | | | | | |
| | 7 | | | | | | | 10 | | | | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | | 3 | | | | | | | 17 | | | | | |
| | 10 | | | | | | | | | | 10 | | | | |
| | 11 | | | | | | | | | | | 10 | | | |
| | 12 | | | | | | | | | | | | | | 10 |
| | 13 | | | | | | | | | | | | | 10 | |
| | 14 | | | | | | | | | | 10 | | | | 20 |

# kNN-luokittelun tulokset ($k = 5$)

Piirrematriisi **A**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 15 | 4 | | | | | | 1 | | | | | | |
| | 2 | 1 | 8 | | | | | | | | | 1 | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | 1 | | | 19 | | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 10 | | | | | | | | |
| | 7 | | | | | | | 10 | | | | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | | | | | | 2 | | | 14 | | | 4 | | |
| | 10 | | | | | | | | | | | | 10 | | |
| | 11 | | | 3 | | | | | | | | 7 | | | |
| | 12 | | | | | | | | | 8 | | | 2 | | |
| | 13 | | | | | | | 10 | | | | | | | |
| | 14 | | | | | | | | 2 | | | | | | 28 |

Piirrematriisi **B**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 16 | 1 | 1 | | | | | 2 | | | | | | |
| | 2 | | 5 | 2 | | | | | | | | 3 | | | |
| | 3 | 1 | | 4 | 5 | | | | | | | | | | |
| | 4 | 1 | | | 18 | | | | | | | 1 | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 9 | | | 1 | | | | | |
| | 7 | | | | | | 1 | 9 | | | | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | | | | | | | | | 17 | | | 3 | | |
| | 10 | | | | | | | | | | | | 9 | | 1 |
| | 11 | | | | | | | | | | | 10 | | | |
| | 12 | | | | | | | | | 5 | 3 | | 1 | | 1 |
| | 13 | | | | | | | 10 | | | | | | | |
| | 14 | | | | | | | | 2 | | | | | | 28 |

Piirrematriisi **C**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 16 | | | 4 | | | | | | | | | | |
| | 2 | | 10 | | | | | | | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | 9 | | | 10 | 1 | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 10 | | | | | | | | |
| | 7 | | | | | | | 10 | | | | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | | 3 | | | | | | | 17 | | | | | |
| | 10 | | | | | | | | | | 10 | | | | |
| | 11 | | | | | | | | | | | 10 | | | |
| | 12 | | | | | | | | | | | | | | 10 |
| | 13 | | | | | | | | | | | | | 10 | |
| | 14 | | | | | | | | | | 10 | | | | 20 |

# kNN-luokittelun tulokset ($k = 7$)

## Piirrematriisi **A**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 15 | 4 | | | | | | 1 | | | | | | |
| | 2 | 3 | 7 | | | | | | | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | 1 | | | 19 | | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 10 | | | | | | | | |
| | 7 | | | | | | | 10 | | | | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | | 1 | | | | 4 | | | 14 | | | 1 | | |
| | 10 | | | | | | | | | | | | 10 | | |
| | 11 | | | | 3 | | | | | | | 7 | | | |
| | 12 | | | | | | | | | 9 | | | 1 | | |
| | 13 | | | | | | | 10 | | | | | | | |
| | 14 | | | | | | | | 2 | | | | | | 28 |

## Piirrematriisi **B**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 15 | 1 | 1 | | | | | 3 | | | | | | |
| | 2 | | 2 | 5 | | | | | | | | 3 | | | |
| | 3 | | | 4 | 6 | | | | | | | | | | |
| | 4 | 1 | | | 19 | | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 9 | | | 1 | | | | | |
| | 7 | | | | | | | 9 | | | 1 | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | | | | | | | | | 20 | | | | | |
| | 10 | | | | | | | | | | | | 9 | | 1 |
| | 11 | | | | 3 | | | | | | | 7 | | | |
| | 12 | | | | | | | | | 7 | 1 | | 1 | | 1 |
| | 13 | | | | | | | 10 | | | | | | | |
| | 14 | | | | | | | | | | 2 | | | | 28 |

## Piirrematriisi **C**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 14 | | | 6 | | | | | | | | | | |
| | 2 | | 10 | | | | | | | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | 8 | | | 10 | 2 | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | 10 | | | | | | | | |
| | 7 | | | | | | | 10 | | | | | | | |
| | 8 | | | | | | | | 20 | | | | | | |
| | 9 | | 3 | | | | | | | 17 | | | | | |
| | 10 | | | | | | | | | | 10 | | | | |
| | 11 | | | | | | | | | | | 10 | | | |
| | 12 | | | | | | | | | | | | | | 10 |
| | 13 | | | | | | | | | | | | | 10 | |
| | 14 | | | | | | | | | | 10 | | | | 20 |

# MLP-verkon luokittelun tulokset (20 ja 30 neuronia)

Piirrematriisi **A**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 10 | | | 8 | 1 | | | | | | | | 1 | |
| | 2 | 1 | | | 7 | 2 | | | | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | | | | 20 | | | | | | | | | | |
| | 5 | | | | | 8 | | | | 2 | | | | | |
| | 6 | | | | | | | | | | | | | | 10 |
| | 7 | | | | | | | | | | | | | | 10 |
| | 8 | | | | 14 | | | | 1 | 5 | | | | | |
| | 9 | | | | | | | | 2 | 18 | | | | | |
| | 10 | | | | | | | | | 5 | | | | | 5 |
| | 11 | | | | 10 | | | | | | | | | | |
| | 12 | | | | | 1 | | | | 5 | | | | 1 | 3 |
| | 13 | | | | | | | | | | | | | 5 | 5 |
| | 14 | | | | | | | | | 4 | | 5 | | 2 | 19 |

Piirrematriisi **B**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 16 | 1 | 1 | 1 | | | | | | | 1 | | | |
| | 2 | 2 | 4 | 1 | 2 | | | | | | | | | | 1 |
| | 3 | 1 | | 6 | 1 | | | | | | | | | | 2 |
| | 4 | | 1 | 1 | 14 | | | | | | | 2 | | | 2 |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | | | | 10 | | | | | |
| | 7 | | | | | | | | | 10 | | | | | |
| | 8 | | | | | | | | 15 | | | | | | 5 |
| | 9 | | | | | | | | | 20 | | | | | |
| | 10 | | | | | | | | | 4 | | | | | 6 |
| | 11 | | | | 10 | | | | | | | | | | |
| | 12 | | | | | | | | | 2 | | | | | 8 |
| | 13 | | | | | | | 3 | | | 2 | | | | 5 |
| | 14 | | | | | | | | 5 | | | | | | 25 |

Piirrematriisi **C**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 10 | | | 10 | | | | | | | | | | |
| | 2 | | 9 | | 1 | | | | | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | 1 | | | 19 | | | | | | | | | | |
| | 5 | | | | 10 | | | | | | | | | | |
| | 6 | | | | 5 | | | | | | | | | | 5 |
| | 7 | | | | | | | | | | | | | | 10 |
| | 8 | | | | | | | | 10 | | | | | | 10 |
| | 9 | | | | | | | | | 14 | | | | | 6 |
| | 10 | | | | | | | | | | | | | | 10 |
| | 11 | | | | | | | | 10 | | | | | | |
| | 12 | | | | | | | | | | | | | | 10 |
| | 13 | | | | | | | | | | | | | | 10 |
| | 14 | | | | | | | | | | | | | | 30 |

# MLP-verkon luokittelun tulokset (50 ja 80 neuronia)

## Piirrematriisi **A**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 12 | 1 | | | | | | 3 | | 2 | | 1 | 1 | |
| | 2 | 6 | 3 | | | | | | 1 | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | | | | 20 | | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | | | | | | | | | 10 |
| | 7 | | | | | | | | | | | | | | 10 |
| | 8 | | | | 3 | | | | 12 | | | | | | 5 |
| | 9 | | 2 | | | | | | | 11 | | | 7 | | |
| | 10 | | | | | | | | | 4 | | | 1 | | 5 |
| | 11 | 1 | | | 8 | | | | 1 | | | | | | |
| | 12 | | | | 1 | 1 | | | | 1 | | | 4 | | 3 |
| | 13 | | | | | | | | | | | | | | 10 |
| | 14 | | | | | | | | | | | | | 2 | 28 |

## Piirrematriisi **B**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 17 | | 1 | 1 | | | | | 1 | | | | | |
| | 2 | 1 | 5 | 1 | 2 | | 1 | | | | | | | | |
| | 3 | | | 5 | 3 | | 1 | | | | 1 | | | | |
| | 4 | | 1 | | 16 | | 1 | | | 1 | 1 | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | | | | 10 | | | | | |
| | 7 | | | | | | | | | | | | | 10 | |
| | 8 | 1 | | | | | | | 13 | | | | | 1 | 5 |
| | 9 | | | | | | | 1 | | 19 | | | | | |
| | 10 | | | | | | | 1 | | 5 | | | | 4 | |
| | 11 | | | | 10 | | | | | | | | | | |
| | 12 | | | | | | | | | 6 | | | 3 | | 1 |
| | 13 | | | | | | | | | | | | | 10 | |
| | 14 | | | 1 | | | | | | | | | | 3 | 26 |

## Piirrematriisi **C**

| | | Luokiteltu luokka | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Todellinen luokka | 1 | 7 | | | 11 | | | | | | | 1 | | | 1 |
| | 2 | | 10 | | | | | | | | | | | | |
| | 3 | | | 10 | | | | | | | | | | | |
| | 4 | | | | 20 | | | | | | | | | | |
| | 5 | | | | | 10 | | | | | | | | | |
| | 6 | | | | | | | | | | | | | | 10 |
| | 7 | | | | | | | | | | | | | | 10 |
| | 8 | | | | | | | | 10 | | | | | | 10 |
| | 9 | | 4 | | | | | | | 3 | | | | | 13 |
| | 10 | | | | | | | | | | | | | | 10 |
| | 11 | | | | 5 | | | | 2 | | | 3 | | | |
| | 12 | | | | | | | | | | | | | | 10 |
| | 13 | | | | | | | | | | | | | | 10 |
| | 14 | | | | | | | | | | | | | | 30 |